



**Universidad Nacional Mayor de San Marcos**

**Universidad del Perú. Decana de América**

**Facultad de Ingeniería de Sistemas e Informática**

**Escuela Profesional de Ingeniería de Sistemas**

**Implementación de una aplicación distribuida basada  
en móviles para el apoyo a la gestión de ventas de una  
empresa de productos lácteos**

**TESIS**

Para optar el Título Profesional de Ingeniero de Sistemas

**AUTOR**

Luis Iván MERA DÁVILA

**ASESOR**

Nora Bertha LA SERNA PALOMINO

Lima, Perú

2019



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

## Referencia bibliográfica

---

Mera, L. (2019). *Implementación de una aplicación distribuida basada en móviles para el apoyo a la gestión de ventas de una empresa de productos lácteos*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

---



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS  
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA  
Escuela Académico Profesional de Ingeniería de Sistemas

Acta de Sustentación de Tesis

Siendo las 7:00pm horas del día 11 de febrero del año 2019 se reunieron los docentes designados como miembros de Jurado de Tesis, presidido por el Mg. Erwin Mac Dowall Reynoso (Presidente), la Mg. Ana María Huayna Dueñas (Miembro) y la Dra. Nora Bertha La Serma Palomino (Miembro Asesor) para la sustentación de la Tesis Intitulada: **"Implementación de una aplicación distribuida basada en móviles para el apoyo a la gestión de ventas de una empresa de productos lácteos"**. del Bachiller: **Luis Iván Mera Dávila**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición de la Tesis, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los Miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el bachiller obtuvo la nota de 17 (En letras) DIECISIETE

A continuación el Presidente del Jurado Mg. Erwin Mac Dowall Reynoso, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 8:00pm horas, se levantó la sesión.

Presidente

Mg. Erwin Mac Dowall Reynoso

Miembro

Mg. Ana María Huayna Dueñas

Miembro Asesor

Dra. Nora Bertha La Serma Palomino



**FICHA CATALOGRÁFICA**

Mera Dávila, Luis Iván

IMPLEMENTACIÓN DE UNA APLICACIÓN DISTRIBUIDA BASADA  
EN MÓVILES PARA EL APOYO A LA GESTIÓN DE VENTAS DE UNA  
EMPRESA DE PRODUCTOS LÁCTEOS

Sistemas, Informática y Sociedad

(Lima, Perú 2019)

Tesis, Facultad de Ingeniería de Sistemas e Informática, Pregrado, Universidad Nacional  
Mayor de San Marcos

Formato 21 x 29.7 cm

**Páginas 124**

## **DEDICATORIA**

El presente trabajo está dedicado a mi querida familia  
por todo su gran amor incondicional.

## **AGRADECIMIENTOS**

Agradezco a Dios por la vida y la salud, a mi familia por todo su amor incondicional, a todos los amigos que me brindaron su apoyo durante y después de mi estadía universitaria y a la Dra. Nora Bertha, La Serna Palomino por su asesoría en este proyecto.

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**IMPLEMENTACIÓN DE UNA APLICACIÓN DISTRIBUIDA BASADA  
EN MÓVILES PARA EL APOYO A LA GESTIÓN DE VENTAS DE UNA  
EMPRESA DE PRODUCTOS LÁCTEOS**

Autor: MERA DÁVILA, Luis Iván  
Asesor: LA SERNA PALOMINO, Nora Bertha  
Título: Tesis para optar el Título de Ingeniero de Sistemas  
Fecha: Noviembre del 2018

---

## **RESUMEN**

El proceso de entrega de una empresa de productos lácteos tiene retardos en el despacho de los pedidos y también se han identificado pérdidas en las ventas. Esto debido al deficiente proceso manual de facturación y control de salidas de los productos terminados. Además, el manejo de la información de los pedidos por parte de la fuerza de ventas muestra ineficiencia en el momento de la toma de éstos, cuando se sale en busca de nuevos clientes y pedidos. Para esto, en este proyecto se implementa una aplicación distribuida que consta de dos componentes que se ejecutarán en distintas plataformas: un componente del lado cliente que se ejecutará en dispositivos móviles y un componente del lado del servidor que brindará servicios distribuidos al componente cliente y que también podrá ser accedido mediante un navegador web para permitir la gestión de ventas. Esta aplicación se realizará siguiendo el estilo arquitectónico web cliente servidor. El componente servidor contendrá una aplicación web que permitirá la gestión de ventas desde un navegador web y una capa de servicios que permitirá el almacenamiento y recuperación de los datos de los pedidos por parte del cliente desde una base de datos centralizada. El componente cliente será una aplicación híbrida para poder ser utilizada como si se tratase de una aplicación nativa y para permitir lograr la independencia del sistema operativo de la aplicación en los dispositivos móviles que se desplieguen. Finalmente se llegó a analizar los resultados de la implantación de esta aplicación. Se logró disminuir considerablemente los tiempos de facturación y de la captación en campo de la información de los clientes y de los pedidos, permitiendo un control más seguro que minimice el margen de error humano como la pérdida de información, cálculos erróneos, extravío y gran almacenamiento de documentos.

**PALABRAS CLAVES:** Aplicación distribuida, web services, aplicaciones móviles híbridas, empresas de productos lácteos, gestión de ventas

**MAJOR NATIONAL UNIVERSITY OF SAN MARCOS**

**FACULTY OF SYSTEMS ENGINEERING AND INFORMATICS  
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING AND INFORMATICS**

**IMPLEMENTATION OF A MOBILE-BASED DISTRIBUTED APPLICATION TO  
SUPPORT THE SALES MANAGEMENT OF A DAIRY PRODUCTS COMPANY**

Author: MERA DÁVILA, Luis Iván  
Adviser: LA SERNA PALOMINO, Nora Bertha  
Title: Thesis, to choose the Professional Title System Engineer  
Date: November 2018

---

**ABSTRACT**

The process of delivering a dairy company has delays in the dispatch of orders and sales losses have also been identified. This is due to the deficient manual process of invoicing and control of outputs of the finished products. In addition, the handling of the information of the orders by the sales force shows inefficiency at the time of taking these when it goes out in search of new customers and orders. For this, in this project a distributed application is implemented that consists of two components that will be executed in different platforms: a client-side component that will run on mobile devices and a server-side component that will provide services distributed to the client component and that It can also be accessed through a web browser to allow sales management. This application will be made following the architectural style web client server. The server component will contain a web application that will allow the management of sales from a web browser and a service layer that will allow the storage and retrieval of the data of the orders by the customer from a centralized database. The client component will be a hybrid application to be used as if it were a native application and to allow the independence of the operating system of the application in the mobile devices that are deployed. Finally, it was possible to analyze the results of the implementation of this application. It was possible to considerably reduce the billing times and the acquisition in the field of customer information and orders, allowing a more secure control that minimizes the margin of human error such as loss of information, erroneous calculations, loss and large storage of documents.

**KEY WORDS:** Distributed application, web services, hybrid mobile applications, dairy products companies, sales management

## ACRÓNIMOS

<b>AMA</b>	American Marketing Association
<b>API</b>	Application Programming Interface
<b>CORBA</b>	Common Object Request Broker Architecture
<b>CORS</b>	Cross-origin resource sharing
<b>CPU</b>	Central Processing Unit
<b>DCOM</b>	Distributed Component Object Model
<b>DEC/RPC</b>	Distributed Computing Environment / Remote Procedure Calls
<b>ERP</b>	Enterprise Resource Planning
<b>GPS</b>	Sistema de Posicionamiento Global
<b>HQL</b>	Hibernate Query Language
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IBM</b>	International Business Machines
<b>IG</b>	Impuesto General a las Ventas
<b>IIS</b>	Internet Information Server
<b>JAX-B</b>	Java Architecture for XML Binding
<b>JAX-RS</b>	Java API extension for RESTful Web services
<b>JDBC</b>	Java Database Connectivity
<b>JPA</b>	Java Persistence API
<b>JSF</b>	Java Server Faces
<b>JSON</b>	JavaScript Object Notation
<b>JSP</b>	Java Server Pages
<b>MVC</b>	Modelo-vista-controlador
<b>NAT</b>	Network Address Translation
<b>ODBC</b>	Open DataBase Connectivity
<b>ORM</b>	Object Relational Mapping
<b>POJO</b>	Plain Old Java Object
<b>RDBMS</b>	Relational Database Management System
<b>RMI</b>	Java Remote Method Invocation
<b>RPC</b>	Remote Procedure Call
<b>RUC</b>	Registro Único de Contribuyentes
<b>RUP</b>	Rational Unified Process
<b>SAE</b>	Sistema de Administración Estudiantil
<b>SAS</b>	Software as a service
<b>SCOP</b>	Sistema de Control de Órdenes de Pedido
<b>SDK</b>	Software Development Kit
<b>SIGV</b>	Sistema de Información para la Gerencia de Ventas
<b>SO</b>	Operating System
<b>SOA</b>	Service-Oriented Architecture
<b>SOAP</b>	Simple Object Access Protocol
<b>SQL</b>	Structured Query Language
<b>SSL</b>	Secure Sockets Layer
<b>SUNAT</b>	Superintendencia Nacional de Aduanas y de Administración Tributaria
<b>UDDI</b>	Universal Description, Discovery, and Integration
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>WEB</b>	World Wide Web
<b>WSDL</b>	Web Services Description Language
<b>XML</b>	Extensible Markup Language

## INDICE

<b>RESUMEN.....</b>	<b>vi</b>
<b>ABSTRACT .....</b>	<b>vii</b>
<b>ACRÓNIMOS.....</b>	<b>viii</b>
<b>INDICE.....</b>	<b>ix</b>
<b>LISTA DE FIGURAS .....</b>	<b>xii</b>
<b>LISTA DE TABLAS.....</b>	<b>xvii</b>
<b>CAPÍTULO I. PLANTEAMIENTO METODOLÓGICO.....</b>	<b>1</b>
<b>1.1 Antecedentes.....</b>	<b>1</b>
<b>1.2 Definición del problema.....</b>	<b>10</b>
<b>1.3 Objetivos .....</b>	<b>11</b>
1.3.1 Objetivo General .....	11
1.3.2 Objetivos Específicos .....	11
<b>1.4 Justificación.....</b>	<b>11</b>
<b>1.5 Alcance .....</b>	<b>12</b>
<b>1.6 Organización de la tesis .....</b>	<b>13</b>
<b>CAPÍTULO II. MARCO TEÓRICO REFERENCIAL.....</b>	<b>14</b>
<b>2.1 Gestión de ventas.....</b>	<b>14</b>
2.1.1 Gestión o administración de ventas .....	14
2.1.2 Planeación de ventas.....	15
2.1.3 Organización de ventas.....	15
2.1.4 Proceso de control de ventas.....	16
<b>2.2 Proceso de ventas.....</b>	<b>16</b>
2.2.1 Proceso de facturación.....	17
2.2.2 Proceso de preparación de pedidos (picking) .....	22
<b>2.3 Sistema de información .....</b>	<b>22</b>

2.3.1 Sistema de información de ventas .....	23
<b>2.4 Aplicaciones Móviles .....</b>	<b>23</b>
2.4.1 Aplicaciones Nativas para Móviles .....	24
2.4.2 Aplicación Web para Móviles.....	25
2.4.3 Aplicaciones Híbridas para Móviles .....	26
<b>2.5. Aplicaciones Distribuidas .....</b>	<b>27</b>
2.5.1 Evolución de las aplicaciones distribuidas .....	29
<b>2.6 Arquitecturas de software .....</b>	<b>30</b>
2.6.1 Arquitectura cliente servidor.....	30
2.6.2 Arquitectura de n capas .....	31
2.6.3 Arquitectura orientada a servicios .....	33
<b>CAPÍTULO III. ESTADO DEL ARTE METODOLÓGICO.....</b>	<b>42</b>
<b>3.1 Trabajos relacionados .....</b>	<b>42</b>
3.1.1 Diseño e implementación de una aplicación distribuida de gestión de inventario para dispositivos móviles. ....	42
3.1.2 Desarrollo de aplicaciones distribuidas online y smart client para dispositivos móviles utilizando .NET, que permitan acceder al sistema de administración estudiantil. ....	43
3.1.3 Desarrollo de un sistema de Información de Registro de pedidos para Ventas usando dispositivos móviles .....	45
<b>3.2 Aplicaciones software .....</b>	<b>46</b>
3.2.1 Vemobile Ventas Express Mobile .....	46
3.2.2 Sevenminds .....	47
3.2.3 Administración de Ventas Móviles (Field Service) .....	48
<b>CAPÍTULO IV. APORTE TEÓRICO .....</b>	<b>51</b>
<b>4.1 Selección y justificación del estilo arquitectónico .....</b>	<b>51</b>
4.1.1 Listado de posibles arquitecturas para la aplicación. ....	51
4.1.2 Definición y Justificación de los Criterios de Evaluación .....	51
4.1.3 Análisis comparativo de las arquitecturas de software .....	52
4.1.4 Arquitectura Seleccionada .....	57
<b>4.2 Adaptación de la arquitectura de software .....</b>	<b>58</b>
<b>CAPÍTULO V. APORTE PRÁCTICO .....</b>	<b>60</b>



<b>5.1 Diseño de la solución.....</b>	<b>60</b>
5.1.1 Requerimientos Funcionales y casos de uso del sistema.....	60
5.1.2 Arquitectura del sistema y diagrama de clases .....	73
5.1.3 Diseño de base de datos.....	80
5.1.4 Diseño de interfaces gráficas .....	81
<b>5.2 Implementación .....</b>	<b>87</b>
5.2.1 Tecnología utilizada para la implementación (Hardware y software).....	87
5.2.2 Frameworks y web services .....	90
5.2.3 Implantación de la solución .....	101
<b><i>CAPÍTULO VI. Conclusiones y trabajos futuros .....</i></b>	<b><i>116</i></b>
<b>6.1 Conclusiones.....</b>	<b>116</b>
<b>6.2 Trabajos futuros.....</b>	<b>117</b>
<b><i>Referencia Bibliográfica.....</i></b>	<b><i>118</i></b>

## LISTA DE FIGURAS

Figura 1.1 Organigrama de la empresa de productos lácteos Artemisa (Fuente: Elaboración propia)...	1
Figura 1.2 Diagrama de flujo del proceso de ventas en Artemisa (Fuente: Elaboración propia) .....	2
Figura 1.3 Diagrama de flujo del subproceso de ventas en campo en Artemisa (Fuente: Elaboración propia).....	3
Figura 1.4 Diagrama de flujo del subproceso de atención de pedidos en Artemisa (Fuente: Elaboración propia).....	5
Figura 1.5 Diagrama de flujo del subproceso de elaboración de productos en Artemisa (Fuente: Elaboración propia).....	6
Figura 1.6 Diagrama de flujo del subproceso de preparación de pedidos en Artemisa (Fuente: Elaboración propia).....	7
Figura 1.7 Diagrama de flujo del subproceso de facturación en Artemisa (Fuente: Elaboración propia) .....	9
Figura 1.8 Diagrama de flujo del subproceso de entrega de pedidos en Artemisa (Fuente: Elaboración propia).....	10
Figura 2.1 Áreas de la administración de ventas (Fuente: (Mercado, 1997)) .....	15
Figura 2.2 Beneficios de la facturación automatizada (Fuente: Adaptado de (Narjiss Afellat, 2011, p. 58) ) .....	18
Figura 2.3 Sistema de información (Fuente:(Stair & Reynolds, 2017)) .....	23
Figura 2.4 Aplicaciones Nativas, Web e Híbridas respectivamente (Fuente: Adaptado de (IBM, 2012)) .....	24
Figura 2.5 Vista de una aplicación web para móviles (Fuente:(Digival, 2016)).....	26
Figura 2.6 Diferentes plataformas en las que se puede ejecutar una aplicación web móvil a través del uso de diferentes frameworks de empaquetado (Fuente: Elaboración propia).....	26
Figura 2.7 Ejemplo de sistema que facilita la ejecución de una aplicación distribuida (Fuente: (U.S. Patent No. 9,838,375 B2, 2017)).....	27
Figura 2.8 Modelo de una forma de ensamblaje de web services xml para aplicaciones distribuidas (Fuente: (Barnett, Kirtland, & Ganapathy, 2003)).....	29
Figura 2.9 Arquitectura cliente servidor con peticiones y respuestas HTTP. Peticiones y respuestas Rest entre cliente y servidor. Y arquitectura cliente servidor web con despliegues en la nube (Fuente: (Saternos, 2014, p. 7)).....	31
Figura 2.10 Diferencias de los conceptos n capas y n niveles (Fuente: (De la Torre et al., 2010, p. 34)) .....	31

Figura 2.11 Ejemplo de una arquitectura de capas (Fuente: (Richards, 2015, p. 6)) .....	32
Figura 2.12 Relación de web services y SOA (Fuente: (Barry & Dick, 2013, p. 30)) .....	34
Figura 2.13 Servicios en un proveedor de servicios (Fuente: (Barry & Dick, 2013, p. 31)) .....	35
Figura 2.14 Ensamblado de servicios dentro de una SOA (Fuente: (Barry & Dick, 2013, p. 30)). ....	35
Figura 2.15 Ejemplos de fuentes en una SOA (Fuente: (Barry & Dick, 2013, p. 32)).....	36
Figura 2.16 Un ejemplo de una vista de arquitectura de servicios de alto nivel para el servicio Cuentas, que representa las partes de la infraestructura circundante utilizadas para cumplir con los requisitos funcionales de todas las capacidades. (Fuente: (Erl, 2016)).....	37
Figura 2.17 La composición del servicio de cuentas desde una perspectiva de modelado (Fuente: (Erl, 2016)).....	37
Figura 2.18 Perspectiva de la arquitectura física de un servicio compuesto (Fuente: (Erl, 2016)). ....	38
Figura 2.19 Ejemplo de una arquitectura de servicio (Fuente: (Erl, 2016)).....	39
Figura 2.20 Elementos de una SOA (Fuente: (Endrei et al., 2004, p. 25)) .....	40
Figura 2.21 Diagrama de componentes de una arquitectura SOA (Fuente: (Krafzig et al., 2004)).....	41
Figura 3.1 Diagrama de componentes utilizado por la aplicación distribuida de gestión de inventario para dispositivos móviles (Fuente: (Víctor, 2011)) .....	43
Figura 3.2 Diagrama de despliegue de la aplicación Web services, Aplicación Online y aplicación Smart Client (Fuente: Adaptado de (Armijos & Pelaéz, 2008)) .....	44
Figura 3.3 Patrón de diseño MVC y vista de implementación de la arquitectura del sistema Información de Registro de pedidos de Ventas usando dispositivos móviles (Fuente: (Balarezo, 2012)) .....	46
Figura 3.4 Interfaz gráfica de usuario de una de las funcionalidades de Ve Mobile en su portal web (Vemobile, 2014). ....	47
Figura 3.5 Interfaz de usuario Sevenminds (Fuente: (Sevenminds, 2016)) .....	48
Figura 3.6 Administración de Ventas Móviles -Field Service (Fuente: (Bpmcapital, 2013)) .....	49
Figura 4.1 Arquitectura del sistema a implementar (Fuente: Elaboración propia) .....	58
Figura 5.1 Actores del sistema (Fuente: Elaboración propia).....	63
Figura 5.2 Diagrama de casos de uso del paquete Facturación y pedidos (Fuente: Elaboración Propia) .....	63
Figura 5.3 Diagrama de casos de uso del paquete Gestión (Fuente: Elaboración Propia) .....	64
Figura 5.4 Diagramas de casos de uso del módulo de reportes (Fuente: Elaboración propia) .....	64
Figura 5.5 Diagrama de casos de uso del módulo Seguridad (Fuente: Elaboración Propia) .....	65

Figura 5.6 Diagrama de casos de uso del módulo Clientes, pedidos y seguridad (Fuente: Elaboración propia).....	65
Figura 5.7 Diagrama de paquetes del sistema (Fuente: Elaboración propia) .....	73
Figura 5.8 Diagrama de componentes del sistema (Fuente: Elaboración propia) .....	73
Figura 5.9 Diagrama de secuencia Caso de uso Gestionar cliente (Fuente: Elaboración propia) .....	74
Figura 5.10 Diagrama de secuencia Caso de uso Gestionar pedidos (Fuente: Elaboración propia).....	75
Figura 5.11 Diagrama de secuencia Caso de uso Emitir comprobante (Fuente: Elaboración propia)..	76
Figura 5.12 Diagrama de despliegue del sistema (Fuente: Elaboración propia) .....	77
Figura 5.13 Diagrama de despliegue ilustrativo del sistema (Fuente elaboración propia) .....	77
Figura 5.14 Diagramas de clases del sistema (Fuente: Elaboración propia) .....	79
Figura 5.15 Esquema de base de datos del sistema (Fuente: Elaboración propia) .....	80
Figura 5.16 Prototipo de interfaz de usuario para el caso de uso de iniciar sesión para el dispositivo móvil (Fuente: Elaboración Propia) .....	81
Figura 5.17 Prototipito de interfaz de usuario del caso de uso registra cliente para el dispositivo móvil (Fuente: Elaboración propia).....	82
Figura 5.18 Prototipito de interfaz de usuario del caso de uso registra pedido para el dispositivo móvil (Fuente: Elaboración propia).....	83
Figura 5.19 Pantalla de inicio del componente web (Fuente: Elaboración propia).....	84
Figura 5.20 Pantalla del caso de uso de gestión de clientes del componente web (Fuente: Elaboración propia).....	84
Figura 5.21 Interfaz del caso de uso gestionar pedido del componente web (Fuente: Elaboración propia).....	85
Figura 5.22 Interfaz del caso de uso registrar pedido del componente web (Fuente: Elaboración propia).....	85
Figura 5.23 Interfaz del caso de uso emitir comprobante del componente web (Fuente: Elaboración propia).....	86
Figura 5.24 Ejemplo de diagrama y página web de Lucidchart(Fuente: (Lucidchart, 2008)) .....	87
Figura 5.25 Sitio web de Creately (Fuente: (Creately, 2008)) .....	88
Figura 5.26 Relaciones de uso entre los componentes de la plataforma de desarrollo del sistema (Fuente: Elaboración propia).....	90
Figura 5.27 Configuración seguida para la habilitación del framework Js y del tema Start de primefaces (Fuente: Elaboración propia) .....	92

Figura 5.28 Arquitectura del framework Jsf con Primefaces (Fuente: Adaptado de: (Ceballos, 2008))	92
Figura 5.29 Actividades realizadas con el ide netbeans para la generación de los archivos necesarios para el manejo del framework Hibernate (Fuente: Elaboración propia)	93
Figura 5.30 El rol de Hibernate en el componente web (Fuente: Adaptado de (Minter et al., 2010))	94
Figura 5.31 Conversiones hechas por JAX-B (Fuente: Elaboración propia)	96
Figura 5.32 Flujo de petición de extremo a extremo de los CORS (Fuente: (Hossain, 2015))	97
Figura 5.33 Respondiendo a una solicitud CORS usando el encabezado Access-Control-Allow-Origin Fuente: (Hossain, 2015))	98
Figura 5.34 Configuración de los CORS desde el lado del servidor en la aplicación para permitir peticiones de diversos orígenes (Fuente: Elaboración propia)	98
Figura 5.35 Esquema de la comunicación de los componentes de la aplicación distribuida (Fuente: Elaboración propia)	99
Figura 5.36 Ejemplo de reporte dinámico hecho con DynamicReports (Fuente: (“DynamicReports”, s. f.))	99
Figura 5.37 Una aplicación simple como una app híbrida trabaja (Fuente: (R. K. Camden, 2016))	100
Figura 5.38 Presentación de Artemisa Lácteos (Fuente:(Artemisa, 2018))	102
Figura 5.39 Algunos productos de Artemisa Lácteos (Fuente: (Artemisa, 2018))	102
Figura 5.40 Gráfico de tiempo en segundos del registro de pedidos con Syslac Web, Syslac Móvil y sin Syslac para los 10 pedidos de la muestra realizados por personas naturales (Fuente: Elaboración propia)	105
Figura 5.41 Gráfico de tiempo en segundos del registro de pedidos con Syslac Web, Syslac Móvil y sin Syslac para los 10 pedidos de la muestra realizados por personas jurídicas (Fuente: Elaboración propia)	106
Figura 5.42 Gráfico de tiempo en segundos de la facturación de pedidos uno a uno sin Syslac y con Syslac para los 10 pedidos de la muestra realizado por personas naturales (Fuente: Elaboración propia)	107
Figura 5.43 Gráfico de tiempo en segundos de la facturación de pedidos de uno en uno sin Syslac y con Syslac para los 10 pedidos de la muestra realizados personas jurídicas (Fuente: Elaboración propia)	108
Figura 5.44 Diagrama de cajas del tiempo en segundos para el registro de pedidos sin Syslac, con Syslac Web y con Syslac Móvil para los 20 pedidos de la muestra (Fuente: Elaboración propia)	110

Figura 5.45 Diagrama de cajas del tiempo en segundos de la facturación de pedidos de uno en uno de la muestra sin Syslac y con Syslac (Fuente: Elaboración propia).....	111
Figura 5.46 Reporte de ventas de los productos en los meses de noviembre del 2017 hasta abril del 2018 con los datos aproximados proveídos por el área de ventas de Artemisa (Fuente: Elaboración propia).....	113
Figura 5.47 Reporte de ventas de los productos en los meses de noviembre del 2017 a abril del 2018 para un cliente de la empresa, obtenido de los datos aproximados proveídos por el área de ventas de Artemisa (Fuente: Elaboración propia) .....	114

## LISTA DE TABLAS

Tabla 2.1 Requisitos de las facturas: (Fuente:(SUNAT, 2016c)) .....	19
Tabla 2.2 Requisitos de las Boletas de Venta (Fuente:(SUNAT, 2016a)) .....	20
Tabla 2.3 Requisitos de las guías de remisión (Fuente:(SUNAT, 2016d)) .....	21
Tabla 2.4 Interfaces de programación para app Nativas (Fuente: (IBM, 2012)).....	24
Tabla 2.5 Interface de programación para app Webs (Fuente: (IBM, 2012)) .....	25
Tabla 2.6 Relación entre web services y SOA (Fuente: Adaptado de (Barry & Dick, 2013, p. 30)) ...	34
Tabla 4.1 Pesos para los Criterios de Evaluación (Fuente: Elaboración propia) .....	56
Tabla 4.2 Notas para los Criterios de Evaluación (Fuente: Elaboración propia) .....	56
Tabla 4.3 Cuadro de Evaluación Comparativo de Arquitecturas de software (Fuente: Elaboración propia).....	57
Tabla 4.4 Notas ponderadas correspondientes a cada arquitectura de software (Fuente: Elaboración propia).....	57
Tabla 5.1 Requerimientos funcionales del módulo Facturación y pedidos (Fuente: Elaboración propia) .....	60
Tabla 5.2 Requerimientos funcionales del módulo Gestión (Fuente: Elaboración propia) .....	61
Tabla 5.3 Requerimientos funcionales del módulo Reportes (Fuente: Elaboración propia) .....	61
Tabla 5.4 Requerimientos funcionales del módulo Seguridad (Fuente: Elaboración propia) .....	61
Tabla 5.5 Requerimientos funcionales del módulo Clientes, Pedidos y Seguridad (Fuente: Elaboración propia).....	62
Tabla 5.6 Requerimientos no funcionales (Fuente: Elaboración propia) .....	62
Tabla 5.7 Especificación del caso de uso Gestionar cliente (Fuente: Elaboración propia) .....	67
Tabla 5.8 Especificación del caso de uso Gestionar pedido (Fuente: Elaboración propia) .....	70
Tabla 5.9 Especificación del caso de uso Emitir comprobante (Fuente: Elaboración propia) .....	72
Tabla 5.10 Anotaciones JAX-RS (Fuente: Adaptado de (Oracle, 2017b)) .....	95
Tabla 5.11 Parámetros proveídos por el api JAX-RS para el uso de web services (Fuente: Adaptado de Oracle, 2017b)) .....	95
Tabla 5.12 Pedidos de personas naturales que se tomaron de muestra de la implantación del sistema (Fuente: Elaboración propia).....	104
Tabla 5.13 Pedidos de personas jurídicas que se tomaron como muestra de la implantación del sistema (Fuente: Elaboración propia).....	104

Tabla 5.14 Tiempo en segundos del registro de pedidos con Syslac Web, con Syslac móvil y Sin Syslac para los 10 pedidos de la muestra realizados por personas naturales (Fuente: Elaboración propia).....	105
Tabla 5.15 Tiempo en segundos del registro de pedidos sin Syslac, con Syslac web y con Syslac móvil para las 10 pedidos de la muestra realizado por personas jurídicas (Fuente: Elaboración propia) .....	106
Tabla 5.16 Tiempo en segundos de la facturación de pedidos uno a uno sin Syslac y con Syslac para los 10 pedidos de la muestra realizados por personas naturales. Donde B, F y G significan boleta, factura y guía de remisión respectivamente (Fuente: Elaboración propia) .....	107
Tabla 5.17 Tiempo en segundos de la facturación de pedidos uno a uno sin Syslac y con Syslac para los 10 pedidos de la muestra realizados por personas jurídicas. Donde B, F y G significan boleta, factura y guía de remisión respectivamente (Fuente: Elaboración propia) .....	108
Tabla 5.18 Tiempo en segundos para la facturación de forma masiva sin Syslac y con Syslac para los 20 pedidos de la muestra (Fuente: Elaboración propia).....	109
Tabla 5.19 Estadísticas de los tiempos en segundos del registro sin Syslac, con Syslac web y con Syslac Móvil de los 20 pedidos de la muestra (Fuente: Elaboración propia) .....	109
Tabla 5.20 Estadísticas de los tiempos en segundos de la facturación uno a uno de los pedidos de la muestra para los escenarios sin Syslac y con Syslac (Fuente: Elaboración propia) .....	110
Tabla 5.21 Tiempos en segundos de la facturación uno a uno vs la facturación masiva de los pedidos de la muestra sin Syslac y con Syslac (Fuente: Elaboración propia) .....	111



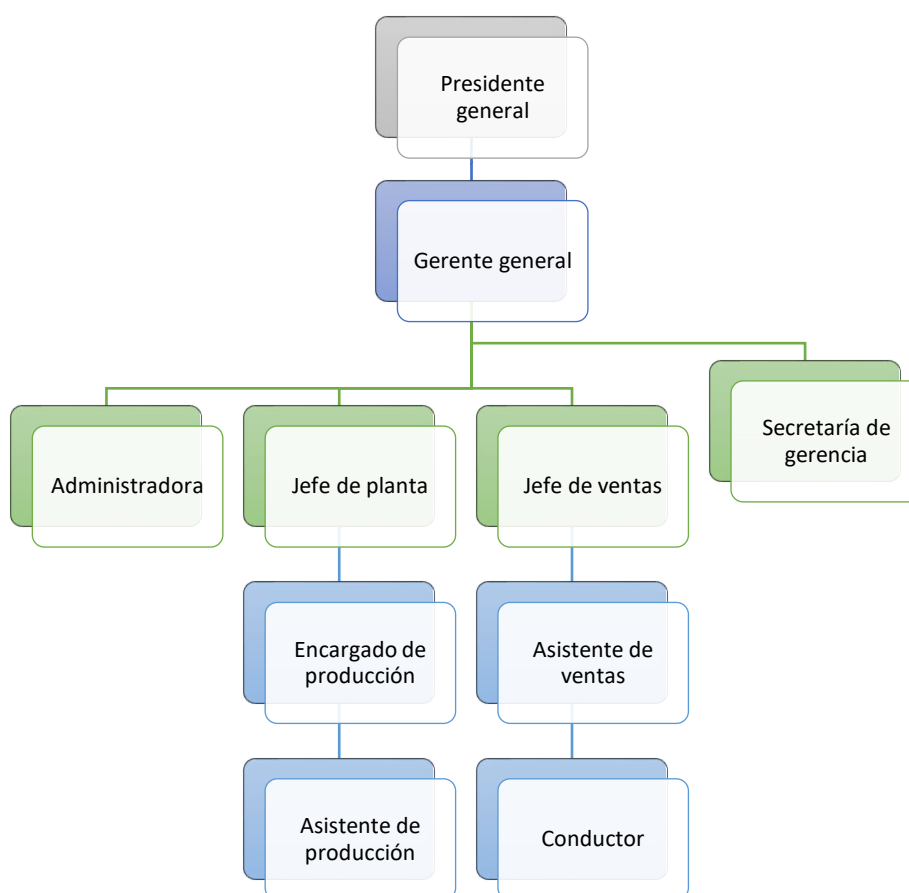
# CAPÍTULO I. PLANTEAMIENTO METODOLÓGICO

## 1.1 Antecedentes

La empresa Artemisa es una empresa dedicada a la producción de productos derivados la leche. Esta empresa cuenta con una planta de elaboración ubicada en el distrito de José Leonardo Ortiz en la provincia de Chiclayo del departamento de Lambayeque.

En la planta de producción se encuentran 4 oficinas: Gerencia General, Oficina de Ventas y Marketing, Administración y la Oficina del Jefe de Planta; acá también está 1 área de producción donde se realiza la elaboración de todos los productos. El área de producción cuenta con las máquinas e implementos con los que se produce. También está conformado por una cámara de frío donde se almacenan los productos terminados, un área de despacho con 1 balanza electrónica donde se pesa los productos para su respectiva entrega y 1 furgoneta para el transporte los pedidos hacia los clientes.

En la siguiente figura se muestra el organigrama de la empresa Artemisa:



**Figura 1.1 Organigrama de la empresa de productos lácteos Artemisa (Fuente: Elaboración propia)**

La empresa Artemisa produce lácteos como son variedades de quesos como: Queso Fresco, Queso Mozzarella, variedades de yogurt como son: Yogurt Frutado, Yogurt de Fresa, Yogurt de durazno, Yogurt de guanábana, Yogurt de mango, Manjar Blanco, Mantequilla y otros productos que son ocasionales y en menor cantidad.

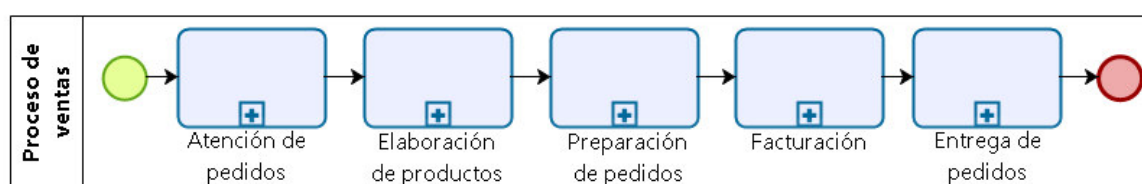
Artemisa cuenta con clientes localizados en los distritos de Chiclayo, José Leonardo Ortiz, La Victoria y Lambayeque. Éstos son personas jurídicas o naturales. La mayoría son frecuentes; los otros son ocasionales. Gran parte utilizan los productos de la empresa como insumos para la producción que ellos ofrecen al mercado diariamente. Por lo que requieren éstos les sean entregados entre un lapso de tiempo fijado.

Los pedidos pueden incluir un determinado peso del producto ya sea queso u otros, un número de litros de yogurt, un número determinado de potes de manjar blanco, un número determinado de potes de mantequilla, un número determinado de litros de leche o la mezcla de algunos o todos estos productos.

La producción cuenta con temporalidad, siendo los meses de verano, los meses de mayor producción donde se contrata más personal y a veces se produce en otro turno en la noche.

La jornada de trabajo empieza a las 7:00 am y termina a las 5:00pm, de lunes a viernes, y el sábado se trabaja de 8:00 am a 1:00pm.

El proceso de ventas cuenta con 5 subprocesos que son: Atención de pedidos, la Elaboración de productos, La Preparación de pedidos, La Facturación y el Despacho de pedidos. En la siguiente figura se muestran los subprocesos del proceso de ventas en Artemisa:



**Figura 1.2 Diagrama de flujo del proceso de ventas en Artemisa**  
(Fuente: Elaboración propia)

La fuerza de ventas de la empresa Artemisa lo constituyen tres personas: una jefa de ventas y marketing y dos asistentes de ventas, todas estas encargadas de salir a brindar degustaciones a los potenciales clientes en distintos distritos de Chiclayo en los establecimientos comerciales de comida, y otros en busca de nuevos clientes para la empresa Artemisa. Para esto utilizan cuadernos para anotar la información de los nuevos clientes que se encuentren y para registrar los pedidos que ellos realicen. Luego es llevado a la oficina de ventas y marketing para su respectiva atención, o estos datos son comunicados telefónicamente.

En la siguiente figura se muestra el diagrama de flujo del subproceso de atención de clientes en campo:

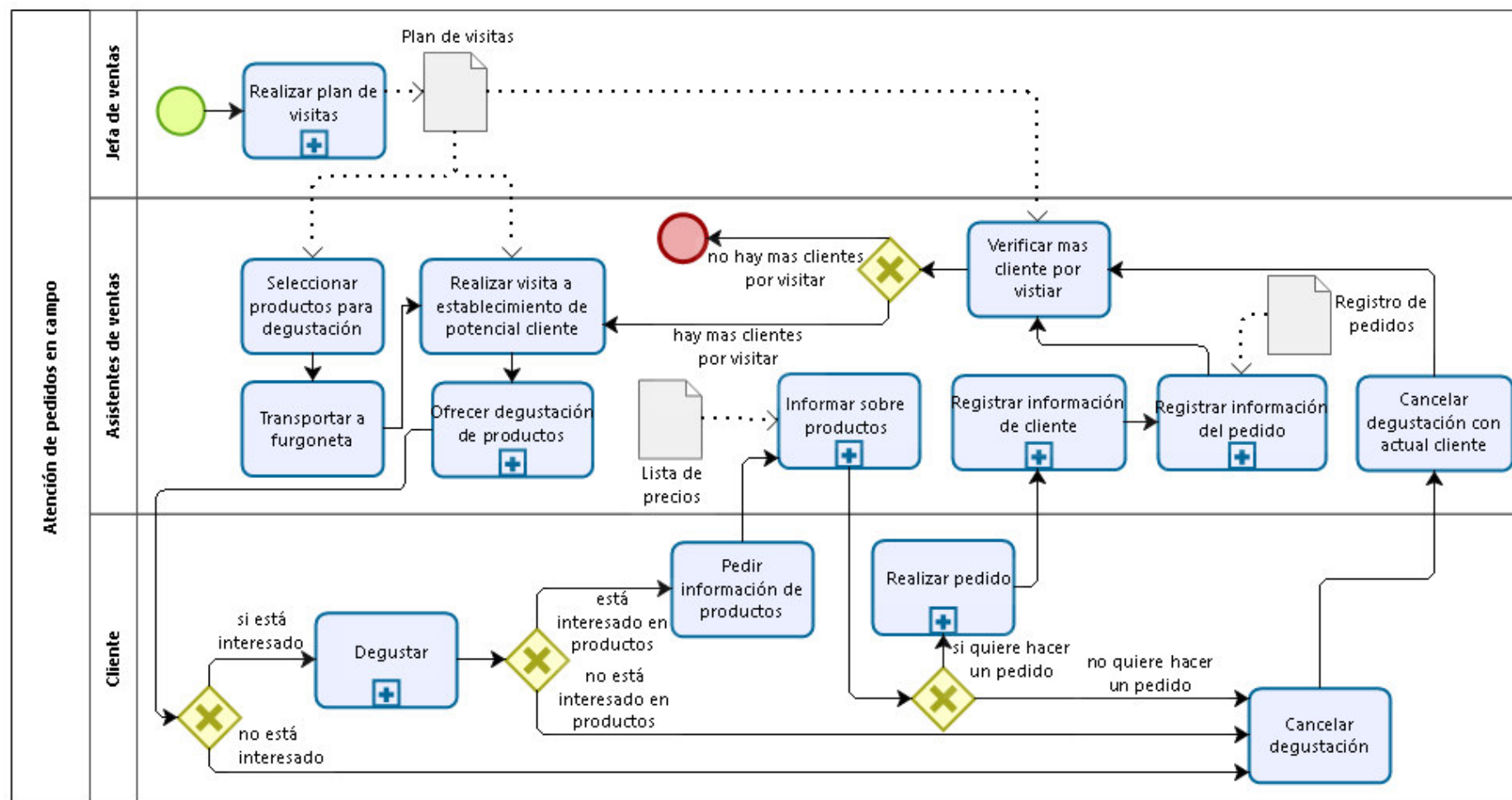
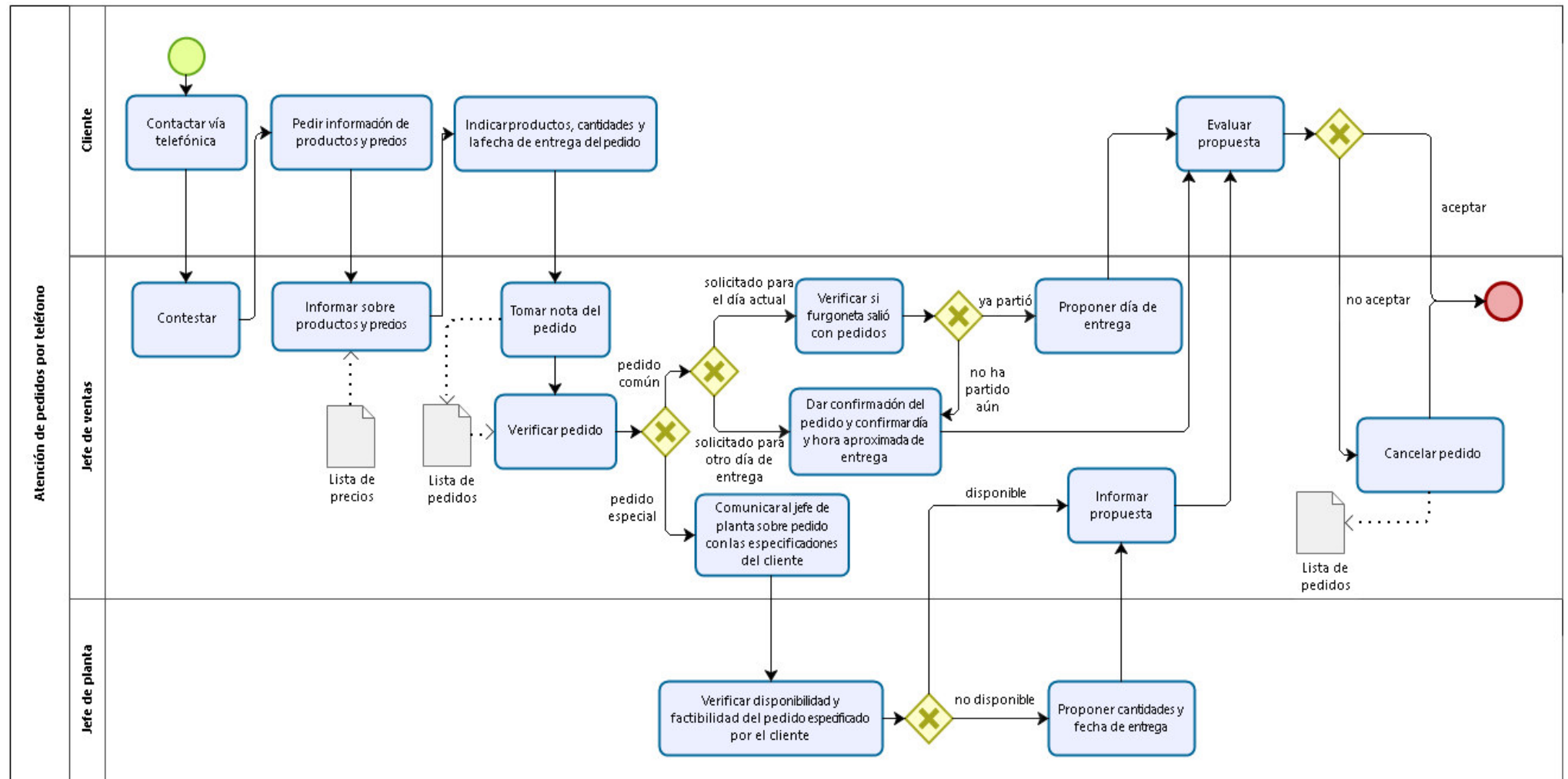


Figura 1.3 Diagrama de flujo del subproceso de ventas en campo en Artemisa (Fuente: Elaboración propia)

El proceso de ventas comienza la atención de los pedidos por parte de la Jefa de Ventas. Los clientes llaman al teléfono fijo o a los celulares de la empresa para realizar sus pedidos. Si el cliente requiere el pedido para ese mismo día, la jefa de ventas averigua si todavía no ha partido a su destino la furgoneta con los pedidos y si las cantidades de productos indicadas en el pedido, están disponibles en stock. Si se cumplen las dos condiciones anteriores, los pedidos son anotados, y pasan a su respectivo despacho; si pasa lo contrario, se anota los datos del pedido, y se pasará para el siguiente día de entrega de pedidos. Todos los pedidos son anotados por la secretaria en el cuaderno de control de pedidos. Todos los pedidos que son atendidos después de la salida de los pedidos son despachados al día siguiente. Luego de recepcionar todos los pedidos del día éstos son pasados a hojas de Excel, luego se las imprime constituyéndose éstas como las hojas de despacho. Al día siguiente estas hojas de despacho son entregadas al ayudante de producción para el respectivo despacho.

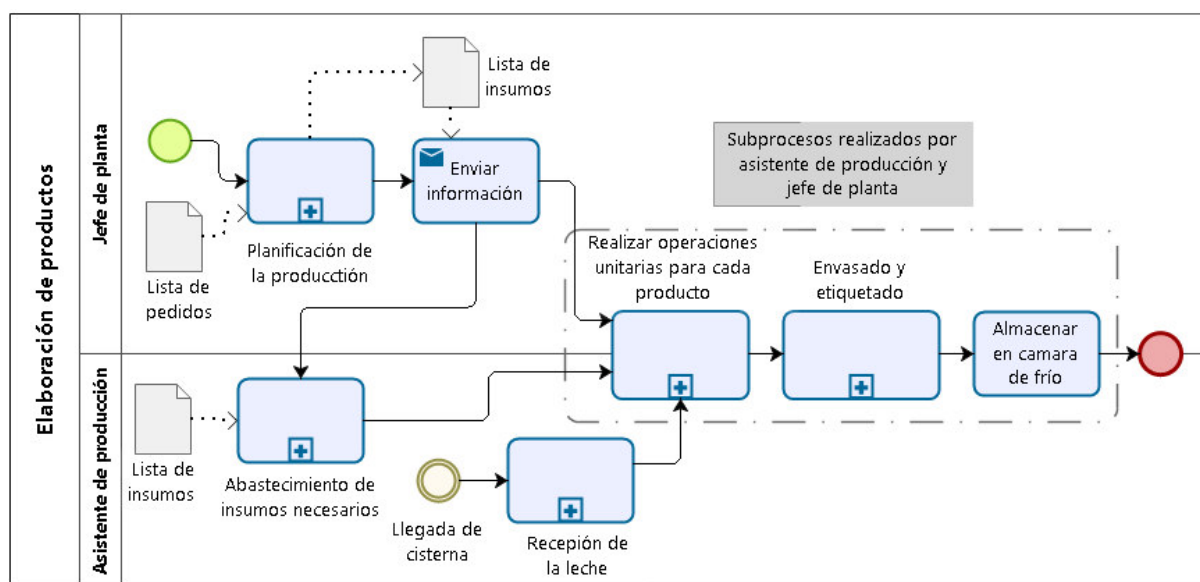
En la siguiente figura se muestra el diagrama de flujo del subproceso de atención de pedidos telefónicamente:



**Figura 1.4 Diagrama de flujo del subproceso de atención de pedidos en Artemisa (Fuente: Elaboración propia)**

La elaboración de los productos se realiza los días lunes, miércoles y viernes. Se produce en función de un plan de producción, dependiendo los pedidos que se realicen, y previendo los pedidos que puedan ser solicitados en los días siguientes. El jefe de planta realiza la planificación de la producción de los diferentes productos; y para eso, envía mediante correo electrónico al asistente de producción y al establo de la empresa una lista con los respectivos insumos y sus cantidades para ser utilizados en la producción. El insumo principal que se usa es la leche. La cisterna que trae la leche llega a la planta y se realiza la entrega de ésta a las 10:00 am aproximadamente. Recibida la entrega de la cisterna de leche el encargado de producción y el ayudante de producción realizan las operaciones unitarias que conforman el proceso de elaboración de cada producto. El envasado de los productos se realiza inmediatamente después de extraer los productos de la cámara de frío. Para el embolsado, en el caso de los quesos, por ejemplo, el queso mozzarella, se embolsa al vacío en una máquina de 6 en 6. Se realiza el envasado de los yogures y de la leche en sus respectivos envases y para los otros productos se utilizan pots. Luego se realiza el etiquetado de la fecha de vencimiento a todos los productos. Los días en que no se produce se realizan las otras etapas de elaboración de productos.

En la siguiente figura se muestra el diagrama de flujo del subproceso de elaboración de productos en la empresa Artemisa:



**Figura 1.5 Diagrama de flujo del subproceso de elaboración de productos en Artemisa (Fuente: Elaboración propia)**

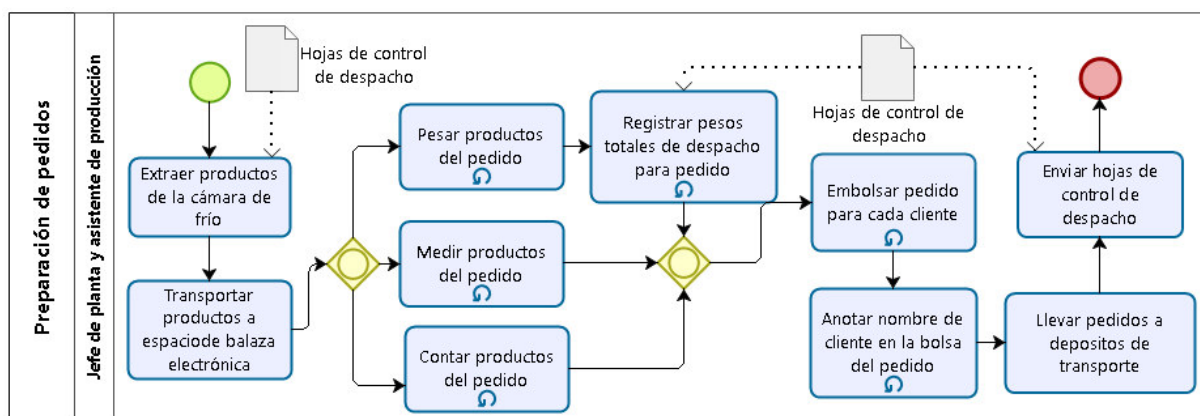
La preparación de pedidos se realiza después del envasado y etiquetado de los productos después del proceso de producción de éstos. En este proceso los participantes son: el jefe de planta, el asistente de producción y la asistente de ventas. Las cantidades de cada producto que

se va a necesitar para la entrega del pedido se obtiene a partir de las hojas de control de despacho. Se extrae de la cámara de frío las cantidades de cada producto y se lleva al lugar de la balanza electrónica para ser pesados (en el caso de los quesos), contados (en el caso de los yogures, pots de manjar blanco, pots de mantequilla y otros) y medidos (cantidad de litros de leche fresca).

En las hojas de control de despachos se apunta el peso a despachar de los productos. Los pedidos pueden especificar una determinada cantidad de kilogramos de queso; pero cada unidad de este producto tiene un peso aproximado (Ejm. La bola de queso Mozzarella pesa 250 g.), por lo cual, se realiza el pesado aproximado a la cantidad pedida (a veces ambas cantidades son iguales y en otras veces lo pesado excede en algo al peso pedido), dicha cantidad debe de ser anotada para su posterior facturación.

Luego de seleccionar, contar, pesar y/o medir los productos de un pedido, éstos se embolsan y el nombre del dueño del pedido es anotado sobre las bolsas correspondientes. De manera paralela, si se reciben pedidos para ser despachados el día actual, éstos son llevados por la jefa de ventas al área de entrega para ser preparados. Cabe mencionar que la oficina de ventas y marketing y el área de preparación de pedidos (ubicada en el área de producción) están alejadas y esto significa un consumo de tiempo para la jefa de ventas que también desatiende muchas veces el teléfono fijo de la empresa dedicada a la atención de pedidos de los clientes y que está ubicado en el área de ventas y marketing. A continuación, el pedido es llevado a un depósito para su transporte, en un lugar donde se concentran todos los pedidos embolsados, para luego ser subidos a la furgoneta para su reparto. Los pedidos seleccionados esperan a que se acabe de procesar manualmente los comprobantes respectivos.

En la siguiente imagen se muestra un diagrama de flujo del subproceso de preparación de pedidos:



**Figura 1.6 Diagrama de flujo del subproceso de preparación de pedidos en Artemisa**  
(Fuente: Elaboración propia)

En la facturación se utilizan formatos impresos. Para ello, se realiza el llenado de la información de las facturas, boletas o guías de remisión remitentes de manera manual, colocan la fecha de la emisión, el nombre del cliente (para los 3 casos), el RUC del cliente (para las facturas y guías de remisión), la dirección de éste (en los 3 casos), fecha del traslado, motivo del transporte y número de comprobante de pago (en el caso de las guías de remisión) los productos que llevan con sus cantidades (en los 3 casos), los subtotales por cada producto (en la factura y boleta), el IGV (en la factura) y el total de todo (en la factura y boleta) usando para esto calculadoras.

El procesamiento manual de comprobantes da cabida a borrones, cambios de tipo de letra o cálculos incorrectos, y ocasiona que los comprobantes tengan que ser anulados y remitidos nuevamente con otro número. Esto constituye un tiempo de retraso en la entrega de los pedidos, y es muy frecuente dentro del proceso de facturación de la empresa Artemisa.

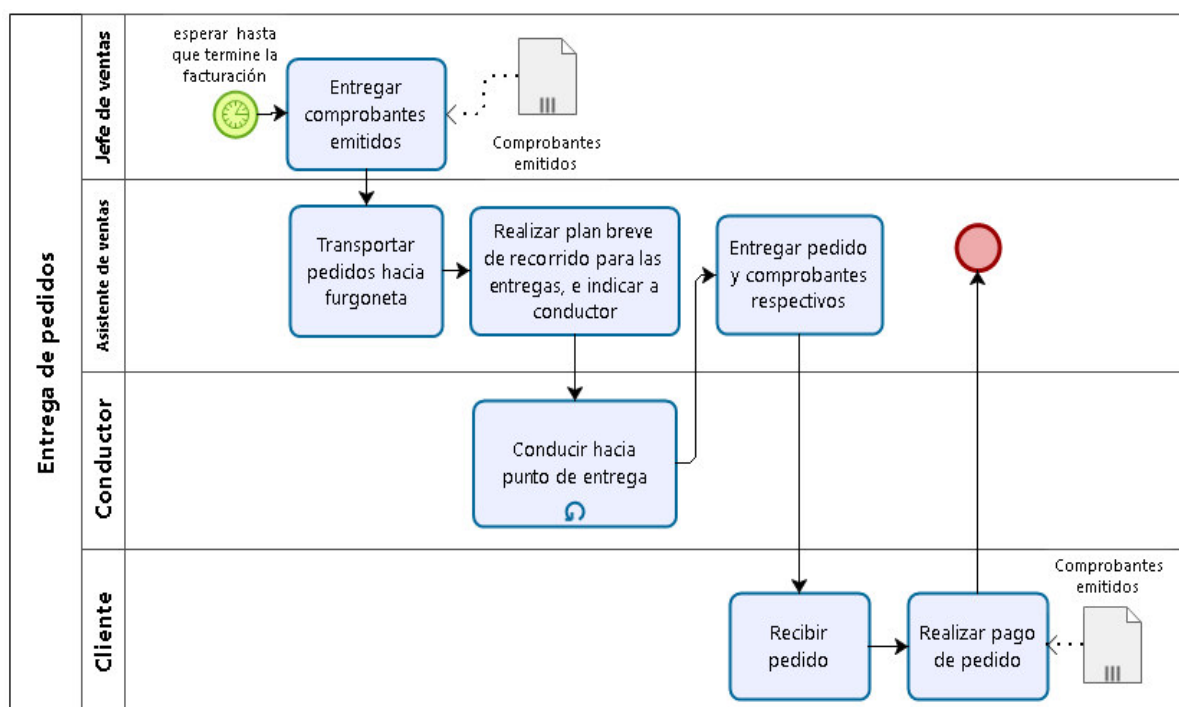
En la siguiente figura se muestra un diagrama de flujo del subproceso de facturación:





La entrega de pedidos se realiza inmediatamente después de la facturación. Cabe recalcar que el inicio de esto depende de la finalización de la facturación, y como se mencionó anteriormente, la facturación es un proceso ineficiente que muchas veces retrasa el inicio del subproceso de entrega de pedidos y esto a la vez causa la entrega tardía de los pedidos, causando el malestar de los clientes que en muchas oportunidades realizan la devolución de los productos. Una vez terminada la facturación, las boletas, facturas, y/o guías de remisión son entregadas a la asistente de ventas que es la responsable de la entrega de pedidos. Se ha fijado un lapso de tiempo para la partida de la furgoneta que es de 9:30 a 10:00 am. Después de la preparación de los pedidos, éstos son llevados a la furgoneta. Se realiza un plan breve del recorrido que se realizará. El vehículo que transporta los pedidos, parte de la planta entre el lapso antes mencionado. Y se realiza el recorrido por los distintos distritos donde se encuentran los clientes, donde se les entrega su respectivo pedido con sus comprobantes. El cliente puede realizar el pago del pedido en este instante o realizarlo después si es que cuenta con crédito. Al terminar la entrega, la furgoneta vuelve a la planta de producción.

En la siguiente figura se muestra un diagrama de flujo del subproceso de entrega de pedidos:



**Figura 1.8 Diagrama de flujo del subproceso de entrega de pedidos en Artemisa**  
(Fuente: Elaboración propia)

## 1.2 Definición del problema

El proceso de entrega de pedidos de esta empresa tiene retardos debido al deficiente procesamiento manual de facturación. Este problema ocasiona frecuentemente la devolución de los productos y la insatisfacción de los clientes.

Además, el registro de los pedidos en campo se realiza de forma manual, lo que origina demora en la atención de los pedidos y en algunos casos la pérdida información.

### **1.3 Objetivos**

#### **1.3.1 Objetivo General**

El objetivo principal de este trabajo de investigación es desarrollar una aplicación distribuida de apoyo a la gestión de ventas que sirva para agilizar este proceso, mejorar el servicio que brinda esta empresa, ayudar al planteamiento de estrategias de marketing, de producción, la toma de decisiones y así hacer frente a la realidad problemática descrita.

#### **1.3.2 Objetivos Específicos**

- Realizar el procedimiento de selección de un estilo arquitectónico de software adecuado para el sistema a implementar.
- Realizar el análisis y diseño del sistema del lado del cliente y del lado del servidor.
- Seleccionar tecnologías de implementación para la codificación del sistema.
- Implementar los componentes cliente y servidor de la aplicación.
- Realizar el análisis de resultados de la implantación del sistema, para verificar el antes y el después de los tiempos de facturación y de registro de pedidos en campo y poder apreciar el impacto de la solución tecnológica desarrollada.

### **1.4 Justificación**

Se seleccionó este proyecto de investigación porque existe en la actualidad una gran demanda de convertir la facturación y el procesamiento de pedidos en actividades más eficaces y eficientes, utilizando un sistema de información para la fuerza de ventas, que permita registrar, calcular, ordenar, gestionar y presentar datos e información de ventas de manera precisa y oportuna.

Las ventajas de este proyecto frente al actual procesamiento manual que conllevarán a una mayor satisfacción del cliente al brindar un mejor servicio, además de permitir un control más seguro que minimice el margen de error humano como la pérdida de información, cálculos erróneos, extravío y gran almacenamiento de documentos son: Contar con un sistema de información compuesto de dos componentes; un componente que permitirá el soporte a la entrega siempre oportuna de pedidos al cliente mediante la reducción de tiempos en los procesos de facturación y, el otro componente, que permitirá el registro de los pedidos sin restricción de tiempo ni espacio. Además, desde el punto de vista técnico proveer un sistema de información reutilizable para futuras implementaciones que se necesiten realizar.

Es relevante porque es indispensable que las empresas siempre brinden su servicio en el lapso de tiempo establecido. Ya que de no ser así el cliente puede cambiar de proveedores, lo cual representaría una pérdida muy importante. Recordemos la razón de existencia de las empresas: Sin clientes no hay empresas.

Los beneficios de esta investigación son la mejora del proceso de ventas de la empresa, reduciendo tiempos empleados en las actividades de captación de pedidos, de los clientes, emisión de comprobantes y la consecuente entrega de productos siempre a tiempo y un mayor soporte a la toma de decisiones en la empresa mediante reportes sobre las ventas. La mejora del proceso de captación de pedidos y de clientes en campo por parte de la fuerza de ventas a través de un móvil, reduciendo los tiempos, la eliminación de los registros manuales y la mejora de la imagen empresarial adaptada a las nuevas tecnologías y la reducción de la carga de trabajo de los empleados.

Los antecedentes narrados anteriormente para la empresa Artemisa demuestran las necesidades de automatización del proceso de ventas de esta pyme. Es posible inferir que estas mismas necesidades pueden afectar a pymes de productos lácteos similares.

### **1.5 Alcance**

El alcance de esta solución ayudará a reducir los tiempos de facturación que permitirán la entrega de pedidos más oportuna.

Se disminuirá el tiempo de registro de los pedidos en campo permitiendo poder enviar la información de los pedidos de manera instantánea sin necesidad de ser transportada y sin el extravío de esta.

El producto final de este trabajo de investigación constará de dos componentes. Uno de los componentes se compondrá de: un módulo de gestión de clientes, un módulo de gestión de pedidos, un módulo de gestión de productos y un módulo generación de comprobantes para la emisión de boletas, facturas, guías de remisión remitente y un módulo de reportes para la emisión de reportes básicos y necesarios para el apoyo en la toma de decisiones de esta empresa.

El otro componente será una aplicación para dispositivos móviles de diferentes plataformas (sistemas operativos, hardware, lenguajes, etc.). El componente cliente tendrá la funcionalidad de captación de información de los clientes y captación de pedidos en campo.

Otras funcionalidades que se podrían considerar para el aplicativo tales como por ejemplo la implementación de la funcionalidad del cliente en modo offline (que permita gestionar la

información localmente sin necesidad de conectividad con el servidor y otras) estará fuera del alcance de este proyecto y será dejado para un futuro trabajo.

## **1.6 Organización de la tesis**

A continuación, se presenta la estructura del documento distribuido en seis capítulos, en cada uno de los cuales se describe con precisión y detalle los conceptos y estudios correspondientes.

En el Capítulo 1 Planteamiento metodológico; se describe los antecedentes, la definición del problema, la importancia que éste tiene, el objetivo principal y los objetivos específicos para afrontar al problema planteado, la justificación y el alcance de esta propuesta de solución.

En el Capítulo 2 Marco Teórico; se muestra cada uno de los conceptos sobre los cuales se fundamentan la gestión de ventas, el proceso de ventas, sistemas de información, aplicaciones móviles, los tipos existentes de éstas: nativas, web e híbridas, el concepto de aplicación distribuida y se presenta un estudio de los estilos arquitectónicos que se podrían adecuar a la construcción de aplicaciones distribuidas como son la arquitectura cliente/servidor, la arquitectura de n capas y la arquitectura orientada a servicios (SOA).

En el Capítulo 3 Estado del Arte Metodológico; se expone, a manera de resumen, tres trabajos relacionados al desarrollo de aplicaciones distribuidas; seguidamente se describe tres aplicaciones de software disponibles en el mercado y que son similares a la aplicación distribuida propuesta en cuanto al funcionamiento y la tecnología.

En el Capítulo 4 Aporte teórico; se describe la comparativa de las tecnologías (arquitecturas) estudiadas en el marco teórico, y seleccionando una de ellas que se adecue al sistema en desarrollo. También se desarrolla la adaptación de la tecnología a este proyecto.

En el Capítulo 5 Aporte práctico; se menciona la metodología utilizada, se presenta el diseño de la solución, herramientas y tecnologías elegidas para la implementación, frameworks, web services, así como los detalles y resultados de la implantación de la solución

En el Capítulo 6 Conclusiones y Trabajos futuros; se presenta las conclusiones obtenidas del trabajo y los posibles trabajos futuros en relación con este proyecto.

## **CAPÍTULO II. MARCO TEÓRICO REFERENCIAL**

En este capítulo se describen los aspectos generales que dan soporte a la investigación. Se abordan conceptos como la Gestión de ventas y los procesos más importantes de ésta: la planificación, la organización y el control. Después, se describe el proceso de ventas, dentro del cual, se aborda al proceso de facturación y el proceso de preparación de pedidos (picking). Seguidamente, se detalla la definición de Sistema de información y sus partes. A continuación, se presentan el concepto de aplicación móvil y de sus tipos. En el siguiente punto, se expone el concepto de aplicación distribuida. Por último, se exponen los estilos arquitectónicos candidatos que pueden ser tomados en cuenta para la implementación de la solución.

### **2.1 Gestión de ventas**

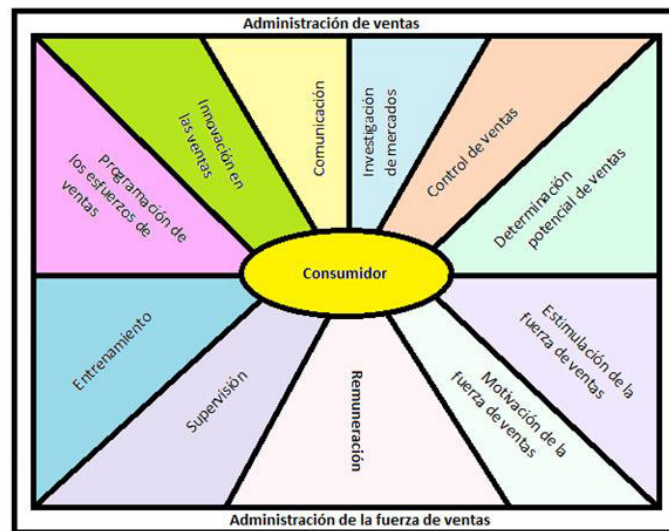
#### **2.1.1 Gestión o administración de ventas**

Según (Asociación de marketing americano, 2014) la gestión de ventas está conformado por 3 procesos. Primero, la formulación del programa estratégico de ventas que implica actividades como desarrollar políticas de administración de cuentas, pronosticar la demanda, el establecimiento de cuotas y presupuestos, la organización y la planificación de las ventas, el despliegue y el enrutamiento. Segundo, la implementación del programa de ventas que incluye actividades de la administración de ventas como la supervisión, selección, reclutamiento, capacitación y motivación de la fuerza de ventas; además el desarrollo de compensación y del incentivo para la fuerza de ventas. Tercero, la evaluación y el control del desempeño que implica el desarrollo y la aplicación de métodos para la supervisión y la evaluación del desempeño de la fuerza de ventas, por ejemplo, tenemos el análisis de comportamiento, costos y de ventas

“Administración de ventas se define como el sistema de administración que mediante el uso de la planeación, organización, dirección, control y coordinación contribuye al incremento de las ventas y sus beneficios” (Mercado, 1997, p. 343).

La gestión de ventas es un proceso organizacional donde se ejecutan actividades que corresponden a la planificación, organización, dirección, control y comunicación dentro de este proceso fundamental de toda empresa con fines de lucro, para poder conseguir los objetivos en cuanto a las salidas de los servicios y o productos que brinda cada empresa. A

sus clientes. Las áreas que comprende la gestión de ventas según (Mercado, 1997), se muestran en la siguiente figura:



**Figura 2.1 Áreas de la administración de ventas (Fuente: (Mercado, 1997))**

### 2.1.2 Planeación de ventas

De acuerdo a (Anderson, Hair Jr., & Bush, 1999), la planeación de ventas permite prever medidas para el futuro con la finalidad de alcanzar las metas y los objetivos planteados, además de ello, para esto, se establecen diagnósticos, pronósticos, políticas, procedimientos, estrategias, tácticas y controles para la gestión de ventas.

Además, el libro de Anderson, Hair y Bush (1999), también señala la importancia de un sistema de información en la gerencia de ventas que posibilite la obtención de la información necesaria del mercado para que ayude a la planificación de las ventas.

### 2.1.3 Organización de ventas

Como se señala en (Anderson et al., 1999), para la organización de ventas se debe tener en cuenta diversos aspectos. Primero, el entendimiento del marco organizacional, consistente en las metas y objetivos, los productos, los tipos de cliente y su distribución, así como la cantidad de vendedores para abastecer el mercado. Segundo, se debe tener en cuenta la segmentación de las actividades de ventas en áreas funcionales tomando como criterio por ejemplo los productos, los tipos de cliente, etc. Tercero, la asignación de vendedores cualificados para los puestos de las actividades de ventas. Cuarto, la asignación de responsabilidades por el desempeño, así como la autoridad necesaria para realizarlas.

(Anderson et al., 1999) también mencionan que la especificación de los puestos de trabajo es fundamental en el establecimiento de la estructura de una organización de ventas más apropiada.

#### **2.1.4 Proceso de control de ventas**

Según (Forsyth, 2002, p. 56) el control de ventas se reduce a la ecuación:

$$A - S = \pm V$$

Donde:

(A) es el rendimiento actual y es comparado con el estándar presente (S), y una acción es considerada para ser tomada si es que existe una diferencia (V) entre los dos.

Según Forsyth, existen tres tipos de estándares:

- Estándares absolutos: Relacionados a objetivos específicos y proveen información importante. (Por ejemplo, el ingreso por ventas anuales, mensuales, trimestrales, etc.).
- Estándares móviles: Son estándares que se miden en el tiempo para evaluar tendencias y para poder pronosticar o prever resultados probables (por ejemplo, el Total Anual Móvil).
- Estándares de diagnóstico: Sirven para mostrar las razones del ¿por qué? De los resultados que se están dando.

Por último, concluye Forsyth, el control de ventas no es solo corregir los malos resultados, sino que también es reforzar los resultados positivos. Por eso que al ver una variación positiva preguntarnos ¿por qué? está pasando eso y trabajar en función de la acción que lo está causando (Forsyth, 2002, p. 57).

### **2.2 Proceso de ventas**

La Asociación de Marketing Americano (AMA) define al proceso de ventas como un conjunto de actividades que son realizadas con el objetivo de lograr la compra de los productos que se ofrecen al mercado. Gracias a la existencia de diversos medios de comunicación las ventas no sólo se realizan personalmente. Por lo general las etapas que conforman este proceso son el análisis de necesidades del cliente, la promoción de las características y beneficios de los



productos ofrecidos y el acuerdo de los precios, entrega, etc. (Asociación de marketing americano, 2014).

Por otra parte, Stanton, Etzel y Walker definen al proceso de venta de la siguiente forma:

“Es una secuencia lógica de cuatro pasos que emprende el vendedor para tratar con un comprador potencial y que tiene por objeto producir alguna reacción deseada en el cliente (usualmente la compra)” (Stanton, Etzel, & Walker, 2004, p. 538).

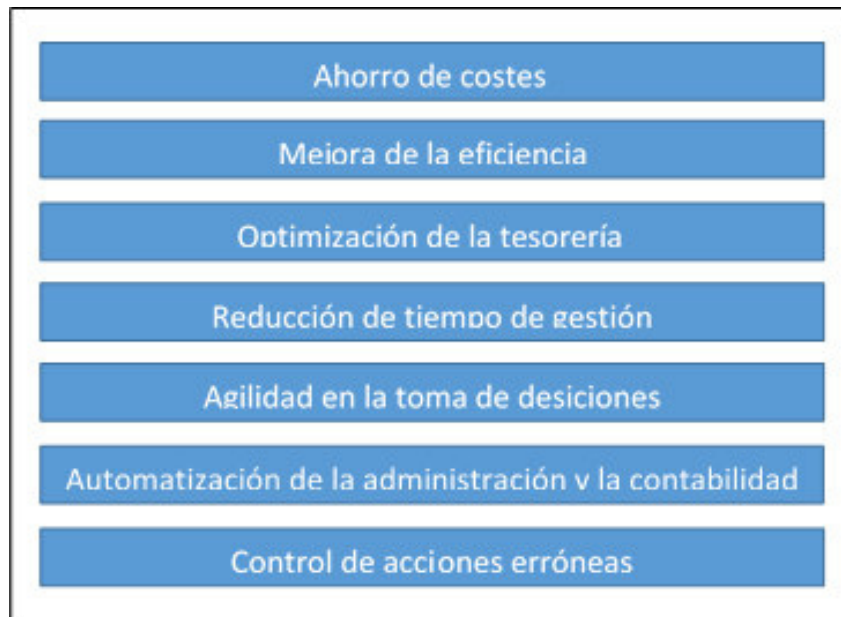
### **2.2.1 Proceso de facturación**

De lo señalado por (Narjiss Afellat, 2011, p. 57), respecto a la facturación electrónica, podemos concluir que la facturación consiste en la emisión de facturas o documentos análogos de un emisor hacia un receptor ya sea por medios electrónicos o a través de medios impresos.

(Narjiss Afellat, 2011) también recalca que el proceso de facturación es un proceso indispensable para las empresas. De forma tradicional estas se realizan manualmente, lo cual, requiere de mucho tiempo y recursos. Es muy importante administrar este proceso de manera ordenada e integra, debido a que contribuye a que la empresa cumpla con sus obligaciones tributarias de forma correcta, además de permitir un mejor control de las entradas y las salidas. Se remarca que la aplicación de nuevas tecnologías para el manejo electrónico de este proceso ayuda mucho a disminuir la carga de trabajo, así como una gestión más ordenada y eficaz de este proceso.

Por tanto, los beneficios de un proceso de facturación automatizado son los ahorros en costes, la reducción de tiempos, la exactitud, disponibilidad que se resumen en eficacia y eficiencia que ayudan a la gestión de ventas de una empresa comercial y por consiguiente a la toma de decisiones de ésta.

En la siguiente figura se muestra los beneficios de un proceso de facturación automatizado.



**Figura 2.2 Beneficios de la facturación automatizada (Fuente: Adaptado de (Narjiss Afellat, 2011, p. 58) )**

### **Comprobantes de pagos (factura, boletas).**

“El comprobante de pago es el documento que acredita la transferencia de bienes, la entrega en uso o la prestación de servicios”(SUNAT, 2016a).

### **Factura**

Según la Superintendencia Nacional de Aduanas y de Administración Tributaria (SUNAT), la factura es el comprobante de pago que se emite en las operaciones entre empresas o personas que necesitan acreditar costo o gasto para efecto tributario, sustentar el pago del IGV por la operación efectuada y poder ejercer, de esta manera, el derecho al crédito fiscal (SUNAT, 2016c).

Por ejemplo, cuando una empresa compra papel y t  ner para sus impresoras debe exigir que le otorguen una factura.

En la siguiente tabla se muestra los requisitos de las facturas seg  n la SUNAT:

INFORMACI��N IMPRESA	INFORMACI��N NO NECESARIAMENTE IMPRESA
Datos de identificaci��n del obligado: <ul style="list-style-type: none"> <li>• Apellidos y nombres, denominaci��n o raz��n social</li> <li>• Adem��s, deben consignar su nombre comercial si lo tuvieran.</li> <li>• Direcci��n del domicilio fiscal y del establecimiento donde est�� localizado el punto de emisi��n.</li> <li>• N��mero de RUC</li> </ul>	Apellidos y nombres, o denominaci��n o raz��n social del adquiriente o usuario
Denominaci��n del Comprobante: “FACTURA”	N��mero de RUC del adquiriente o usuario
Numeraci��n: Serie y n��mero correlativo	Bien vendido o cedido en uso, descripci��n o tipo de servicio prestado, indicando la cantidad, unidad de medida, n��mero de serie y/o n��mero de motor.
Datos de la imprenta que realizo la impresi��n: <ul style="list-style-type: none"> <li>• N��mero de RUC</li> <li>• Fecha de impresi��n</li> </ul>	Precios unitarios de los bienes vendidos o importe de la cesi��n de servicios prestados
N��mero de autorizaci��n de impresi��n otorgado por la SUNAT: que debe estar junto con los datos de la empresa gr��fica	Valor de la venta o importe del servicio prestado, sin incluir los tributos que afectan la operaci��n ni otros cargos.
Destino del original y las copias: <ul style="list-style-type: none"> <li>• En el original: ADQUIRIENTE O USUARIO</li> <li>• En la primera copia: EMISOR</li> <li>• En la segunda copia: SUNAT</li> <li>• En las copias se consignar�� la leyenda “COPIA SIN DERECHO A CR��DITO FISCAL DEL IGV”</li> </ul>	Monto discriminado de los tributos que gravan la operaci��n y otros cargos adicionales, indicando el nombre del tributo y/o concepto y la tasa correspondiente.
	N��mero de la gu��a de remisi��n u otro documento relacionado con la operaci��n.
	Fecha de emisi��n
	C��digo de autorizaci��n emitido por el Sistema de Control de ��rdenes de Pedido (SCOP), en la venta de combustibles l��quidos y otros productos derivados de los hidrocarburos.

**Tabla 2.1 Requisitos de las facturas: (Fuente:(SUNAT, 2016c))**

### Boleta de ventas

Seg  n la SUNAT estos son comprobantes de pago que se emiten: “En las operaciones con consumidores o usuarios finales, [...], no permitir  n ejercer derecho al cr  dito fiscal ni podr  n sustentar costo y gasto para efecto tributario, salvo en los casos que la Ley lo permita”(SUNAT, 2016a).

La siguiente tabla muestra los requisitos de una boleta de venta según la Superintendencia Nacional de Aduanas y de Administración Tributaria (SUNAT):

INFORMACIÓN IMPRESA	INFORMACIÓN NO NECESARIAMENTE IMPRESA
Datos de identificación del obligado: <ul style="list-style-type: none"> <li>• Apellidos y nombres, o denominación o razón social</li> <li>• Además, deben consignar su nombre comercial si lo tuvieran.</li> <li>• Dirección del domicilio fiscal y del establecimiento donde esté localizado el punto de emisión.</li> <li>• Número de RUC.</li> </ul>	Bien vendido o cedido en uso, tipo de servicio prestado, número de serie y/o número de motor si se trata de un bien identificable.
Denominación del Comprobante:  “BOLETA DE VENTA”	Importe de la venta o del servicio prestado.
Numeración Serie y número correlativo	Fecha de emisión
Datos de la imprenta que realizó la impresión: Número de RUC Fecha de impresión	Cuando el importe de la boleta de venta supere los S/ 700.00, será obligatorio consignar los datos de identificación del adquiriente o usuario <ul style="list-style-type: none"> <li>• Apellidos y nombres</li> <li>• Número de documento de identidad</li> </ul>
Número de autorización de impresión otorgado por la SUNAT: que debe estar junto con los datos de la empresa gráfica	
Destino del original y las copias: En el original: EMISOR En la primera copia: ADQUIRIENTE O USUARIO	

**Tabla 2.2 Requisitos de las Boletas de Venta (Fuente:(SUNAT, 2016a))**

### Guías de remisión

Según la SUNAT “Las Guías de Remisión son documentos que sustentan el traslado de bienes entre distintas direcciones” (SUNAT, 2016d).

Además, la SUNAT señala que existen modalidades para la emisión de los tipos de guía de remisión: remitente y transportista. Si el transporte es privado se debe de emitir sólo guía de remisión remitente y si es transporte público se emitirá una guía de remisión transportista y una guía de remisión remitente(SUNAT, 2016d).

En la tabla siguiente se muestra los requisitos para una guía de remisión remitente según la Superintendencia Nacional de Aduanas y de Administración Tributaria (SUNAT):

INFORMACIÓN IMPRESA		INFORMACIÓN NO NECESARIAMENTE IMPRESA	
<b>Datos de identificación del remitente:</b> <ul style="list-style-type: none"> <li>• Apellidos y nombres, denominación o razón social</li> <li>• Se debe consignar el nombre comercial si lo tuvieran.</li> <li>• Domicilio fiscal y del establecimiento donde esté localizado el punto de emisión. Podrá consignarse la totalidad de direcciones de los establecimientos que posee el contribuyente.</li> <li>• Número de RUC</li> </ul>		<b>Dirección del punto de partida:</b> <ul style="list-style-type: none"> <li>• Excepto si el mismo coincide con el punto de emisión del documento</li> </ul>	
<b>Denominación del Comprobante:</b> “GUÍA DE REMISIÓN-REMITENTE”		<b>Dirección del punto de llegada</b>	
<b>Numeración</b> Serie y número correlativo		<b>Fecha de emisión del “Comprobante de Retención”</b>	
<b>Motivo del traslado:</b>		<b>Datos de identificación del destinatario</b>	
<ul style="list-style-type: none"> <li>• Venta</li> <li>• Compra</li> <li>• Devolución</li> <li>• Consignación</li> <li>• Importación</li> <li>• Exportación</li> </ul>	<ul style="list-style-type: none"> <li>• Venta sujeta a confirmación</li> <li>• Traslado entre establecimientos</li> <li>• de la misma empresa</li> <li>• Traslado de bienes para transformación</li> <li>• Recojo de bienes</li> <li>• Traslado por emisor itinerante</li> <li>• Traslado zona primaria</li> <li>• Venta con entrega a terceros</li> <li>• Otras no incluida en los puntos anteriores.</li> </ul>	<ul style="list-style-type: none"> <li>• Apellidos y nombres, denominación o razón social</li> <li>• Número de RUC, salvo que no esté obligado a tenerlo, en cuyo caso se deberá consignar el tipo y número de documento de identidad.</li> <li>• En caso el destinatario sea el mismo remitente se consignará: “EL REMITENTE”</li> </ul>	
<b>Datos de la imprenta que realizó la impresión:</b> <ul style="list-style-type: none"> <li>• Número de RUC</li> <li>• Fecha de impresión</li> </ul>		<b>Datos de identificación de la unidad de transporte y conductor</b> <ul style="list-style-type: none"> <li>• Marca y número de placa</li> <li>• Número de licencia de conducir</li> </ul>	
<b>Número de autorización de impresión otorgado por la SUNAT:</b> que debe estar junto con los datos de la empresa gráfica		<b>Datos del bien transportado</b> <ul style="list-style-type: none"> <li>• Descripción detallada</li> <li>• Cantidad y peso total</li> <li>• Unidad de medida</li> </ul>	
<b>Destino del original y las copias:</b> <ul style="list-style-type: none"> <li>• En el original: DESTINATARIO</li> <li>• En la primera copia: REMITENTE</li> <li>• En la segunda copia: SUNAT</li> </ul>		<b>Fecha de inicio del traslado</b>	

**Tabla 2.3 Requisitos de las guías de remisión (Fuente:(SUNAT, 2016d))**

### **2.2.2 Proceso de preparación de pedidos (picking)**

“Se refiere a la actividad de la recolección de la mezcla correcta de los productos en sus correctas cantidades desde el área de almacenamiento teniendo en cuenta las cantidades solicitadas en la orden del pedido” (Huber, s. f.).

De acuerdo a (Mauleón, 2003) el proceso de recogida es la obtención y acondicionamiento de los productos contenidos en los detalles de pedidos de los clientes por parte del personal encargado.

## **2.3 Sistema de información**

“Un sistema de información es un conjunto de elementos interrelacionados que recaban, procesan, almacenan y distribuyen datos e información, y, además, proporcionan mecanismos de retroalimentación para alcanzar un objetivo” (Stair & Reynolds, 2017, p. 10).

(Stair & Reynolds, 2017) nos describen las 3 partes de un sistema de información. Primero, la entrada, que son las acciones que realiza el sistema para la captación de los datos que se deben ingresar al sistema para realizar el procesamiento de la información. Por ejemplo, para el cálculo del sueldo de un trabajador se tiene que haber ingresado primeramente las horas trabajadas por el trabajador en el periodo correspondiente. Segundo, el procesamiento, que consiste en la transformación de datos de entrada en salidas a través de realización de cálculos, el empleo de algoritmos, la grabación de datos de forma persistente, etc. Por ejemplo, para un sistema de identificación de identidad mediante preguntas, primero se tiene que captar los datos de identificación de la persona, el sistema recupera información registrada anteriormente de diversas fuentes, en función de esto arma preguntas que se preguntarán, las cuales, pueden involucrar alternativas aleatorias y una alternativa correcta siguiendo un determinado criterio. De acuerdo a los que conteste el evaluado el sistema realizará el cálculo del puntaje obtenido en dicha evaluación y confirmará la identidad de la persona o la rechazará según sea el caso. Esta transacción es grabada en una base de datos para temas de auditoría y de cobranzas. Tercero, la salida consiste en la información útil que se presenta al usuario, como por ejemplo boletas de pago de trabajadores, reportes de ventas en un periodo de tiempo, etc. En muchos casos los sistemas se pueden integrar de forma que la salida forme la entrada de otro para complementar el procesamiento de la información de los diversos procesos del negocio, por ejemplo, un sistema de pedidos puede complementarse con un sistema de facturación.

En la siguiente figura se muestran los componentes de un sistema de información:



**Figura 2.3 Sistema de información (Fuente:(Stair & Reynolds, 2017))**

### **2.3.1 Sistema de información de ventas**

Según (Laudon & Laudon, 2012, p. 338), un sistema de información de ventas es aquel que da soporte a una de las principales funciones del negocio: las ventas.

Además, el “Proceso de ventas y marketing, comprende procesamiento de pedidos, cotizaciones, contratos, configuración de productos, precios, facturación, verificación de créditos, administración de incentivos y comisiones, y planificación de ventas” (Laudon & Laudon, 2012, p. 338). Por tanto, podemos concluir que un sistema de ventas es aquel que da soporte a cualquiera de estas actividades.

## **2.4 Aplicaciones Móviles**

Según (Santiago, Trinaldo, Kamijo, & Fernández, 2015):

Se denomina aplicación móvil o app es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros.

En la siguiente figura se muestran los tipos de aplicaciones móviles existentes:



**Figura 2.4 Aplicaciones Nativas, Web e Híbridas respectivamente**  
(Fuente: Adaptado de (IBM, 2012))

### 2.4.1 Aplicaciones Nativas para Móviles

De acuerdo a (IBM, 2012) las aplicaciones nativas están compuestas de archivos binarios que se instalan en el dispositivo móvil. El instalador de la aplicación se puede descargar de las tiendas de aplicaciones móviles respectivas al sistema operativo. Cuando se instala dicha aplicación esta se ejecuta directamente sobre el sistema operativo, sin necesidad de ningún otro intermediario y además tiene acceso a las APIs expuestas por el SO.

En cuanto al desarrollo de aplicaciones nativas IBM menciona que se necesita del “kit de desarrollo de software” (SDK) del sistema operativo móvil ya que el SDK es específico de la plataforma y cada sistema operativo móvil viene con sus propias herramientas únicas (IBM, 2012).

La siguiente tabla presenta las diferentes herramientas, lenguajes, formatos y canales de distribución asociados con los principales sistemas operativos móviles.

	<i>Apple iOS</i>	<i>Android</i>	<i>Blackberry OS</i>	<i>Windows Phone</i>
<b>Lenguajes</b>	<i>Objective-C, C, C++</i>	<i>Java (algo de C, C++)</i>	<i>Java</i>	<i>C#, VB.NET y más.</i>
<b>Herramientas</b>	<i>Xcode</i>	<i>Android SDK</i>	<i>BB Java Eclipse Plug-in</i>	<i>Visual Studio, herramientas de desarrollo de Windows Phone.</i>
<b>Formato del empaquetado</b>	<i>.app</i>	<i>.apk</i>	<i>.cod</i>	<i>.xap</i>
<b>App stores</b>	<i>Apple App Store</i>	<i>Google Play</i>	<i>Blackberry App World</i>	<i>Windows Phone Marketplace</i>

**Tabla 2.4 Interfaces de programación para app Nativas (Fuente: (IBM, 2012))**



### 2.4.2 Aplicación Web para Móviles

Una aplicación web para móviles es una aplicación web con formato para smartphones y tabletas, y se accede a través del navegador web del dispositivo móvil. Como una aplicación web tradicional, una aplicación web móvil está construido con tres tecnologías núcleos: HTML (define el texto estático e imágenes), CSS (define el estilo y la presentación) y JavaScript (define las interacciones y animaciones). Dado que las aplicaciones web son basadas en el navegador, no dependen de la plataforma donde se ejecuten, siendo capaces de funcionar en cualquier teléfono inteligente o tableta. Una aplicación web móvil normalmente es descargada de un servidor web central cada vez que se ejecuta, aunque la aplicación construida utilizando HTML5 también se puede ejecutar en el dispositivo móvil para su uso sin conexión (Lionbridge, 2012).

En la tabla siguiente se muestra las características de las aplicaciones web móviles puras y de los sitios web móviles puros, tomando como criterios de comparación a las herramientas para su construcción, el modo de ejecución, la experiencia de uso del usuario y su performance:

<b><i>Características</i></b>	<b><i>Apps web móviles puras</i></b>	<b><i>Sitios web móviles puros</i></b>
<b><i>Herramientas y conocimiento</i></b>	<i>Escrito enteramente en HTML, CSS y JavaScript</i>	<i>Escrito enteramente en HTML, CSS y JavaScript</i>
<b><i>Ejecución</i></b>	<i>Acceso directo "Instalado", lanzado como una aplicación nativa.</i>	<i>Alcanzada por navegar a un sitio web a través de un localizador uniforme de recursos (URL)</i>
<b><i>Experiencia de usuario</i></b>	<i>Interfaz de usuario táctil amable, e interactiva</i>	<i>Interfaz de usuario de navegación entre las páginas que muestran datos estáticos</i>
<b><i>Performance</i></b>	<i>La lógica de la interfaz de usuario reside localmente, por lo que la aplicación es ágil y accesible sin conexión a internet</i>	<i>Todo el código ejecutado desde un servidor, lo que resulta en el rendimiento que depende de la red</i>

**Tabla 2.5 Interface de programación para app Webs (Fuente: (IBM, 2012))**

Una ventaja de este tipo de aplicaciones, es que solo basta acceder a ellas a través de un navegador web instalado en el dispositivo, logrando así la independencia del sistema operativo sobre el cual se ejecute. Para el desarrollador no es necesario aprender lenguajes de programación específicos, asociados a cada sistema operativo, como lo es en el caso de las aplicaciones nativas que cada plataforma tiene asociado un lenguaje en particular: las aplicaciones sobre Android están asociadas a Java o como el caso de los Ios de Apple que se usa C o C++.

La figura siguiente muestra una aplicación web que es accedida desde diversos dispositivos como tablets, smartphones y computadoras, por ejemplo.



**Figura 2.5 Vista de una aplicación web para móviles (Fuente:(Digival, 2016))**

### **2.4.3 Aplicaciones Híbridas para Móviles**

Las aplicaciones híbridas se desarrollan con tecnologías web para múltiples plataformas móviles y también acceden a las APIs expuestas por el sistema operativo usando un navegador interno como mediador. Existen diversas soluciones que proveen una interfaz JavaScript que es igual para todos los sistemas operativos móviles por ejemplo PhoneGap, ApacheCordova, etc.

La figura siguiente muestra las diferentes plataformas en las que puede ejecutarse una aplicación gracias al uso de diferentes frameworks para el empaquetamiento de aplicaciones web html5 para convertirlas en híbridas.



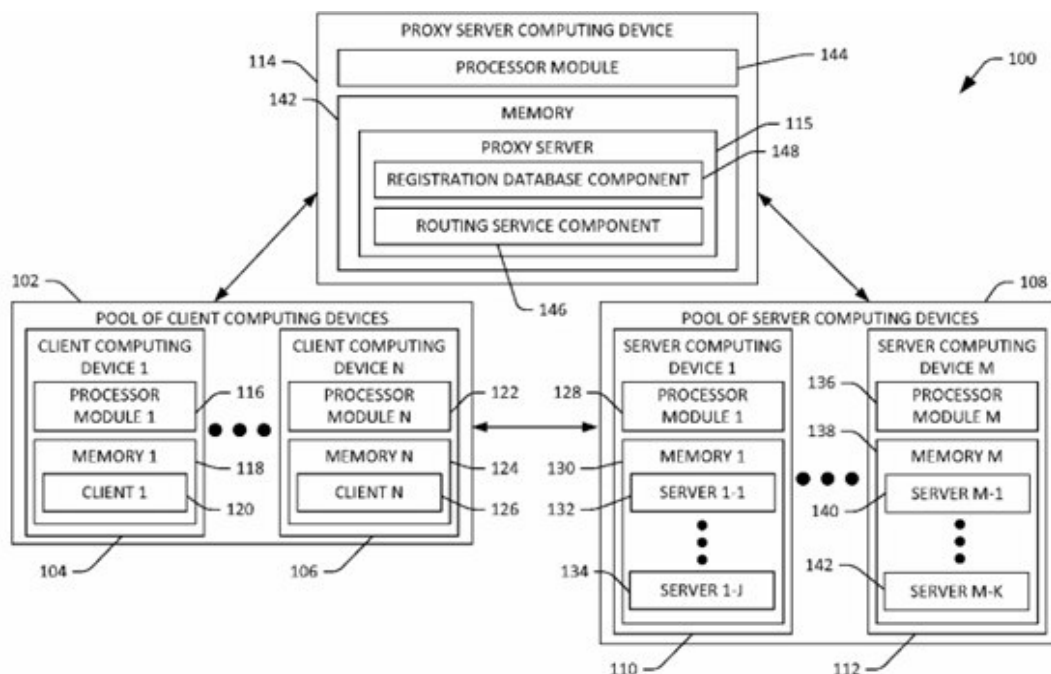
**Figura 2.6 Diferentes plataformas en las que se puede ejecutar una aplicación web móvil a través del uso de diferentes frameworks de empaquetado (Fuente: Elaboración propia)**

## 2.5. Aplicaciones Distribuidas

En (U.S. Patent No. 9,838,375 B2, 2017) se nos hablan de las aplicaciones distribuidas claramente:

Una aplicación distribuida es un software que se ejecuta en dos o más computadoras dentro de una red, en la que estos ordenadores pueden comunicarse uno con el otro por medio de una red. Generalmente, en un entorno cliente-servidor, una aplicación distribuida tiene dos partes: Un "front end" que es ejecutado sobre un dispositivo informático cliente; y un "back-end" que se ejecuta en otros dispositivos de computación, tales como un dispositivo informático servidor (s). La aplicación distribuida puede hacer uso de una Interfaz de programación de aplicaciones (API) para proporcionar una interfaz común para las comunicaciones entre diferentes instancias de ejecución, y se puede usar para proporcionar servicios, características e información de una manera que pueda ser entendido por desarrolladores de la aplicación y que pueda ser utilizado por diferentes instancias en ejecución de la aplicación.

En la siguiente figura se muestra un sistema que facilita la ejecución de una aplicación distribuida que utiliza un API RESTlike:



**Figura 2.7 Ejemplo de sistema que facilita la ejecución de una aplicación distribuida (Fuente: (U.S. Patent No. 9,838,375 B2, 2017))**

El sistema que incluye un conjunto de dispositivos de computación cliente (102) que comprende una pluralidad de dispositivos informáticos (104-106). Los dispositivos de computación cliente (104-106) pueden incluir dispositivos de computación de escritorio, dispositivos portátiles de computación, netbooks, ultrabooks, tabletas, phablets, teléfonos móviles, wearables (relojes inteligentes o smartwatches, gafas inteligentes o smartglasses, zapatillas de deportes con GPS incorporado, pulseras que monitorizan el estado de salud entre otros). El sistema (100) incluye además un conjunto de dispositivos de computación de servidor (108) que comprende una pluralidad de dispositivos de computación servidor (110-112). Además, el sistema (100) incluye un dispositivo informático servidor proxy (114). Los dispositivos informáticos cliente (104-106), dispositivos de computación de servidor (110-112), y el dispositivo informático servidor proxy (114) están en comunicación uno con el otro por medio de conexiones de red respectivas y adecuadas. Además, los canales de comunicaciones pueden establecerse directamente entre dispositivos informáticos cliente en el grupo de dispositivos de computación cliente 102 o entre un dispositivo de computación cliente en el grupo de dispositivos de computación cliente 102 y un dispositivo de computación de servidor en el grupo de dispositivos de computación 108 para facilitar las comunicaciones entre ellas (U.S. Patent No. 9,838,375 B2, 2017).

En un mundo cada vez más conectado, las empresas y los usuarios demandan aplicaciones distribuidas, las aplicaciones distribuidas permiten a los objetos hablar a través de límites del proceso. A menudo, aplicaciones distribuidas también deben cumplir con los siguientes objetivos:

- Establece comunicación entre objetos que se ejecutan en diferentes dominios de aplicación y procesos ya sea en el mismo ordenador o a través de Internet.
- Permite integración de aplicaciones empresariales mediante el establecimiento de comunicación entre objetos que se ejecutan en arquitecturas heterogéneas.
- Permite la disponibilidad de las aplicaciones asegurándose de que las partes de la aplicación continúan ejecutándose incluso si algunos componentes están ocupados o han fallado.
- Proporcionar mayor seguridad y escalabilidad al dividir la aplicación en varias capas (o niveles) (Kalani & Kalani, 2003).

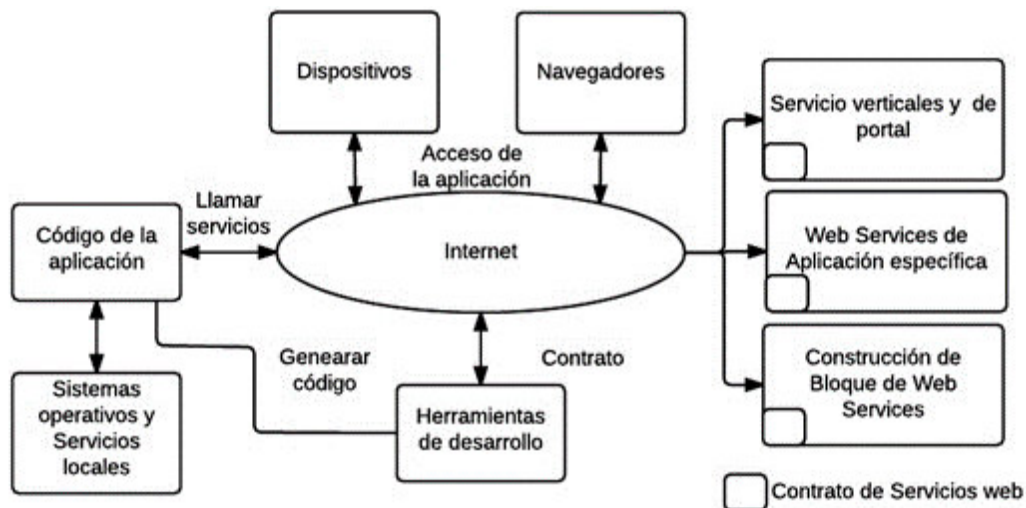
Como hemos visto las aplicaciones distribuidas son diferentes aplicaciones o componentes corriendo sobre diferentes sistemas operativos contruidos con diferentes modelos de objetos

que pueden usar diferentes lenguajes de programación y que se integran dentro de un robusto sistema o aplicación para el soporte de diversos procesos del negocio o programas de investigación científica.

Uno de los más importantes desafíos dentro de las implementaciones de las aplicaciones distribuidas es la integración de la aplicación.

Además, los desarrolladores de aplicaciones crecientemente quieren y necesitan alcanzar más allá de los entornos cliente servidor para acceder a funcionalidades en sistemas remotos a nivel web o sobre la internet y que son muy diferentes dentro del diseño e implementación, y los cuales son propietarios y se gestiona para otras localizaciones.

En la figura siguiente se presenta un modelo que ilustra como los servicios web xml pueden ser ensamblados para crear aplicaciones web distribuidas completamente funcionales.



**Figura 2.8 Modelo de una forma de ensamblaje de web services xml para aplicaciones distribuidas (Fuente: (Barnett, Kirtland, & Ganapathy, 2003))**

### 2.5.1 Evolución de las aplicaciones distribuidas

Se han hecho varios esfuerzos para diseñar frameworks para el desarrollo de aplicaciones distribuidas. Existen frameworks conocidos tales como: DEC/RPC (1984), DCOM (1997), CORBA (1991), RMI (1995) de Java. Algunas de éstas implementaciones están ampliamente desplegadas en las empresas (Kalani & Kalani, 2003; Krakowiak, 2009).

## **2.6 Arquitecturas de software**

### **2.6.1 Arquitectura cliente servidor.**

“Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios (clientes) y procesos que responden a estos servicios (servidores). Estos dos tipos de procesos pueden ejecutarse en el mismo procesador o en distintos” (Mora, 2002, p. 40).

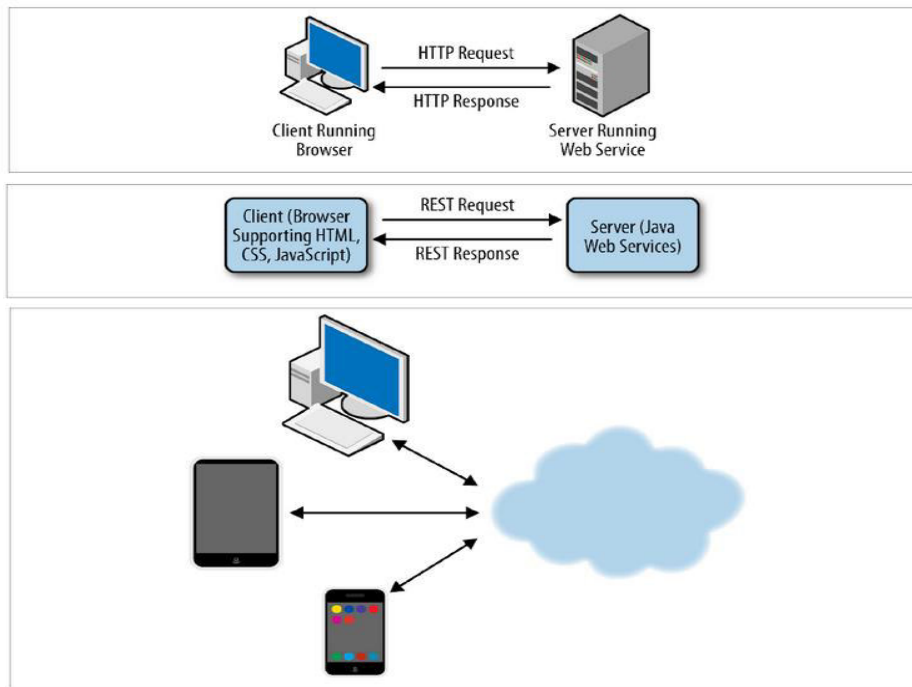
De acuerdo a (Mora, 2002, p. 44), los sistemas que siguen esta arquitectura presentan las siguientes características. Primero, una composición de dos partes. La parte cliente que hace de interfaz entre el usuario y el sistema y la parte del servidor que se comunica con las bases de datos, impresoras, modems, etc. Segundo, los requerimientos de recursos son diferentes tanto en el cliente como en el servidor por ejemplo en cuanto a memoria, velocidad, capacidad de los discos duros, etc. Tercero, ambas partes suelen ser distintos en cuanto al sistema operativo, lenguaje en el cual son desarrollados y memoria; donde el cliente interactúa a través de una API o RPC con el servidor. Cuarto, la mayoría de la parte cliente consiste de una interfaz gráfica de usuario.

Según (Saternos, 2014, p. 6) la naturaleza de la web no ha cambiado ya que sigue siendo de la misma arquitectura cliente servidor donde existen servidores que sirven documentos HTML mediante el protocolo HTTP.

Saternos también nos habla respecto a la arquitectura web cliente-servidor que se puede concluir que es una arquitectura basada en la web (que es una arquitectura cliente servidor) con la diferencia existente en que varían un poco respecto al protocolo. Por ejemplo, REST se utiliza para las arquitecturas web cliente servidor, y que se desarrolló en función a HTTP definiendo sobre éstas restricciones para poder construir una aplicación confiable, escalable con un diseño elegante, simple y mantenible (Saternos, 2014, p. 6).

(Saternos, 2014, p. 7) enfatiza los desafíos en la web actual debido a que debemos considerar múltiples dispositivos y despliegues en la nube.

La siguiente figura muestra ejemplos de arquitectura cliente servidor, cliente servidor web y del entorno actual de la aplicación del estilo arquitectónico cliente servidor a través con despliegues de servidores en la nube:

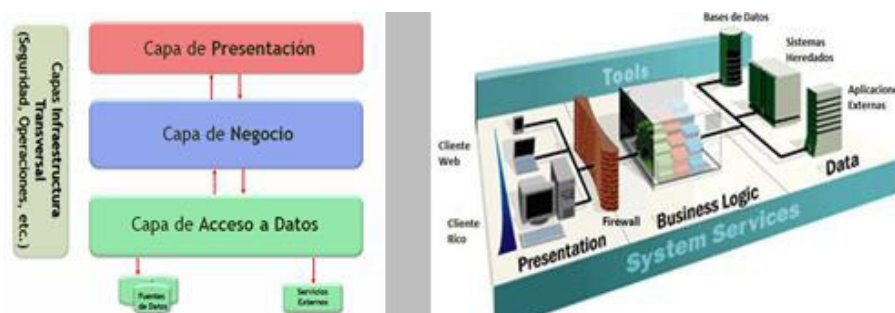


**Figura 2.9 Arquitectura cliente servidor con peticiones y respuestas HTTP. Peticiones y respuestas Rest entre cliente y servidor. Y arquitectura cliente servidor web con despliegues en la nube (Fuente: (Saternos, 2014, p. 7))**

### 2.6.2 Arquitectura de n capas

(De la Torre, Ramos, & Calvo, 2010, p. 33) resalta la diferencia entre capas y niveles. Las capas consisten en separación lógica de la aplicación, en cambio los niveles describen la separación física de los componentes de dicho sistema.

En la figura siguiente se muestra tanto un esquema 3-Tier y un esquema N-Layer donde se pueden observar las diferencias comentadas (distribución lógica vs. distribución física):



**Figura 2.10 Diferencias de los conceptos n capas y n niveles (Fuente: (De la Torre et al., 2010, p. 34))**

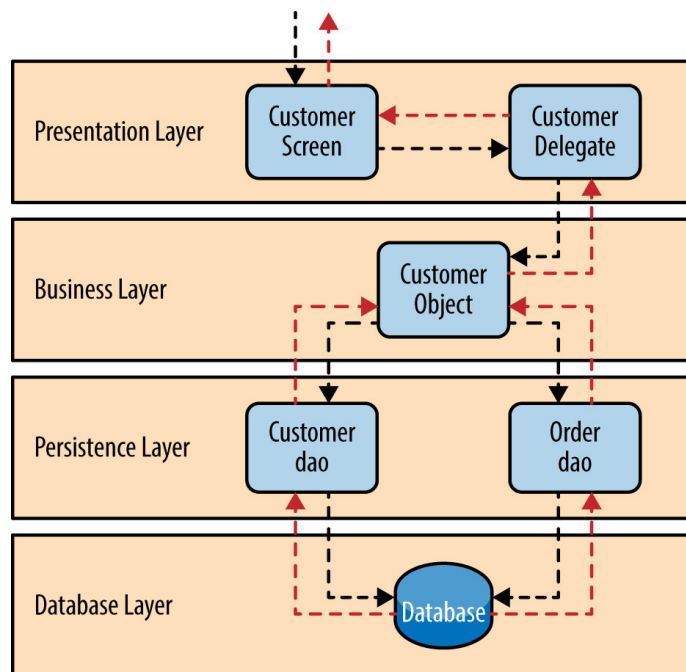
Según (Richards, 2015, p. 1) este patrón de arquitectura es el más común. Además, es un estándar para la mayoría de aplicaciones empresariales de Java, y por tanto las más conocida por arquitectos, analistas y programadores.

Los componentes son organizados dentro de capas horizontales con un específico rol para cada una. Por ejemplo, presentación, lógica del negocio, persistencia y base de datos. En algunos casos la capa de negocio y la capa de persistencia son combinadas dentro de una única capa del negocio, particularmente cuando la lógica de persistencia (Ejemplo: SQL, o HQL) es embebida dentro de los componentes de la capa del negocio (Richards, 2015, p. 1).

El concepto clave de esta arquitectura el “aislamiento de capas”. Este concepto significa que los cambios que se hagan en una capa de la arquitectura generalmente no impactarán o afectarán componentes de otras capas (Richards, 2015, p. 3).

(Richards, 2015, p. 4) cita un ejemplo para entender la importancia del “aislamiento de capas”. Al considerar el esfuerzo para convertir el framework de presentación de JSP (Java Server Pages) a JSF (Java Server Faces) la capa del negocio permanece intacta e independiente de la capa de presentación.

En la figura a continuación se muestra el ejemplo de una arquitectura en capas.



**Figura 2.11 Ejemplo de una arquitectura de capas (Fuente: (Richards, 2015, p. 6))**

La pantalla del cliente es responsable de aceptar la solicitud y mostrar la información del cliente. No sabe dónde están los datos, cómo se recuperan ni cuántas tablas de bases de datos deben ser consultas para obtener los datos. Una vez que la pantalla del cliente recibe una solicitud para obtener información del cliente para un individuo en particular, luego envía esa solicitud al módulo delegado del cliente. Este módulo es responsable de saber qué módulos en la capa de negocio pueden procesar esa solicitud y también cómo llegar a ese



módulo y qué datos necesita (el contrato). El objeto del cliente en la capa de negocio es responsable de agregar toda la información que necesita la solicitud comercial (en este caso, para obtener información del cliente). Este módulo llama al módulo de dao (objeto de acceso a datos) del cliente en la capa de persistencia para obtener datos del cliente, y también al módulo de pedido para obtener información sobre el pedido. Estos módulos a su vez ejecutan sentencias de SQL para recuperar los datos correspondientes y volver a pasarlos al objeto del cliente en la capa de negocio. Una vez que el objeto del cliente recibe los datos, agrega los datos y los vuelve a pasar al delegado del cliente, que luego los pasa a la pantalla del cliente para presentarlos al usuario (Richards, 2015, p. 6).

### **2.6.3 Arquitectura orientada a servicios**

D. Barry y D. Dick nos definen a la arquitectura orientada a servicios de la siguiente forma:

“La arquitectura orientada a servicios es una forma de diseñar, implementar y ensamblar servicios para dar soporte o automatizar funciones de negocios. Varios servicios web se pueden usar para conectar servicios” (Barry & Dick, 2013, p. 15).

De acuerdo a D. Barry y D. Dick debemos de describir dos conceptos muy importantes dentro de SOA:

#### ***Servicios***

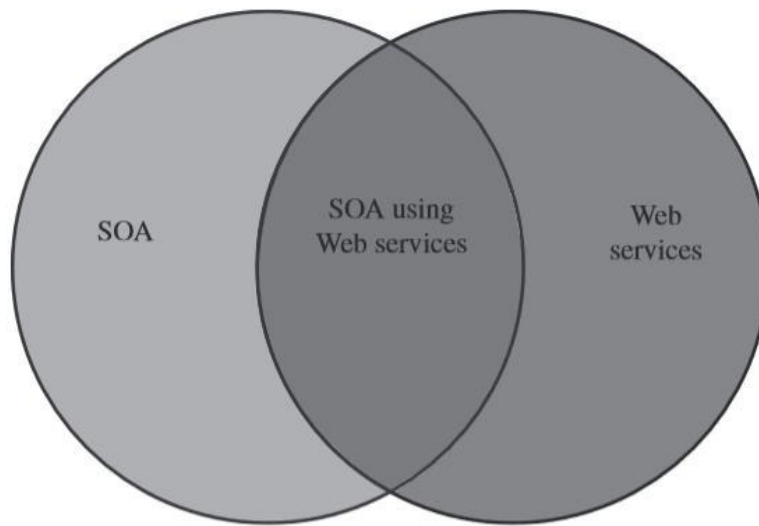
Un servicio es software y hardware. Uno o más servicios admiten o automatizan una función de negocio. En la mayoría de los casos, la intención es que un servicio se pueda usar de múltiples maneras (a menudo denominado reutilización). Hay dos tipos de servicios: atómico y compuesto. Un servicio atómico es una función bien definida y autónoma que no depende del contexto o estado de otros servicios. Un servicio compuesto es un conjunto de servicios compuestos atómicos u otros. Un servicio dentro de un servicio compuesto puede depender del contexto o estado de otro servicio que también esté dentro del mismo servicio compuesto (Barry & Dick, 2013, p. 17).

#### ***Conexiones***

“Los servicios web proporcionan los medios para conectar servicios” (Barry & Dick, 2013, p. 18).

Barry y Dick también mencionan la existencia de múltiples especificaciones que se pueden usar para implementar Web Services tales como SOAP / WSDL sin UDDI, REST, XML y JSON (Barry & Dick, 2013, p. 19).

La Figura siguiente usa un diagrama de Venn para ilustrar la relación entre SOA y los servicios web:



**Figura 2.12 Relación de web services y SOA (Fuente: (Barry & Dick, 2013, p. 30))**

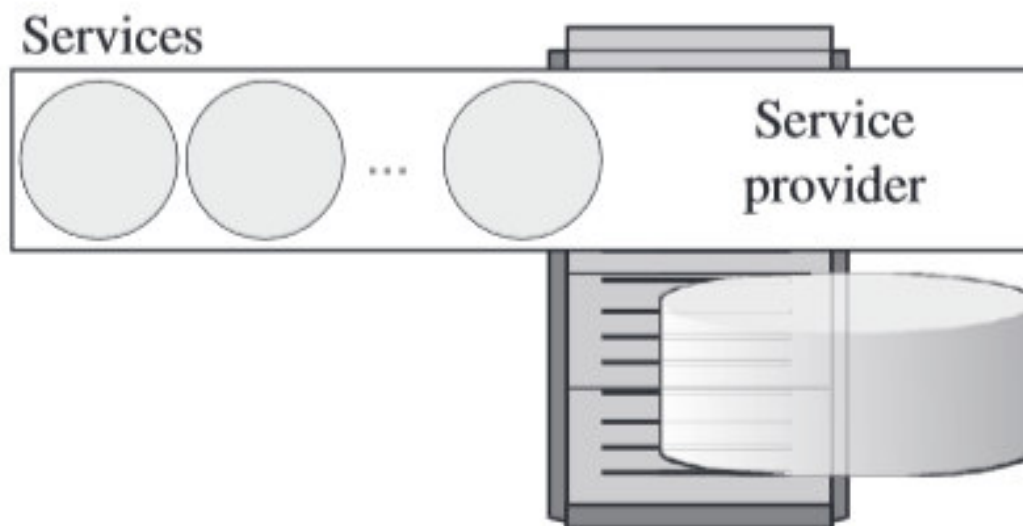
Las partes de estos conjuntos representan:

$SOA \cap \text{Web Services}$	SOA utilizando servicios web para conexiones
$\text{Web Services} - (SOA \cap \text{Web Services})$	Los servicios web se pueden usar para las conexiones, pero las conexiones por sí solas no constituyen una SOA
$SOA - (SOA \cap \text{Web Services})$	SOA indica que una SOA puede usar servicios web, así como otras conexiones que no sean servicios web (las especificaciones originales de CORBA y DCOM son ejemplos).

**Tabla 2.6 Relación entre web services y SOA (Fuente: Adaptado de (Barry & Dick, 2013, p. 30))**

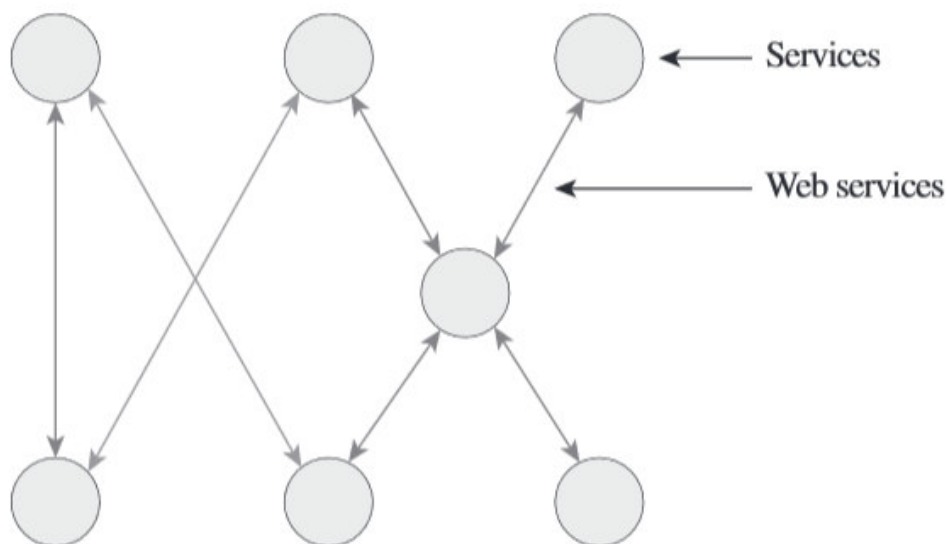
### **SOA**

Según (Barry & Dick, 2013, p. 31), cualquier proveedor de servicios podría proporcionar múltiples servicios. Los servicios se ejecutan en código en un sistema informático subyacente, que proporciona computación, así como acceso y actualizaciones a los datos almacenados. Los pequeños círculos representan los servicios múltiples en la figura siguiente:



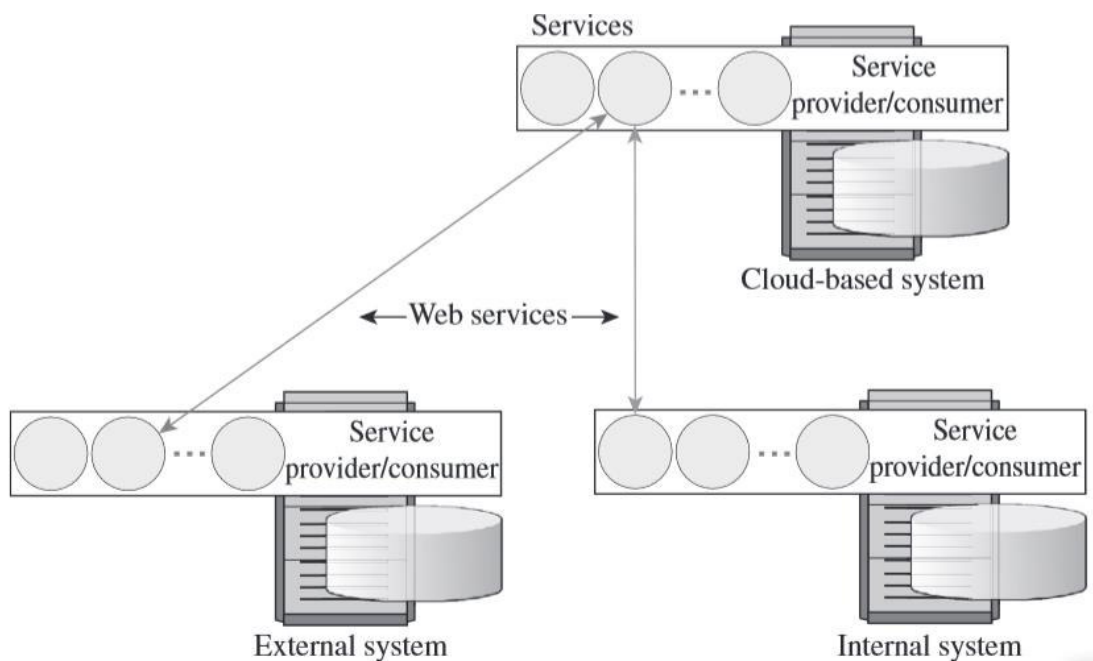
**Figura 2.13 Servicios en un proveedor de servicios (Fuente: (Barry & Dick, 2013, p. 31))**

Los servicios se ensamblan para soportar o automatizar las funciones del negocio. Una SOA es reasentada mediante la siguiente figura, donde los servicios web se usan para conectar los servicios en una SOA (Barry & Dick, 2013, p. 32):



**Figura 2.14 Ensamblado de servicios dentro de una SOA (Fuente: (Barry & Dick, 2013, p. 30)).**

Los servicios en una SOA pueden provenir de cualquier proveedor de servicios y también puede ser consumidores de servicios. Por lo tanto, en una SOA determinada, los servicios pueden ser de sistemas internos junto con cualquier cantidad de sistemas externos accesibles en cualquier lugar de Internet (Barry & Dick, 2013, p. 32):



**Figura 2.15 Ejemplos de fuentes en una SOA (Fuente: (Barry & Dick, 2013, p. 32)).**

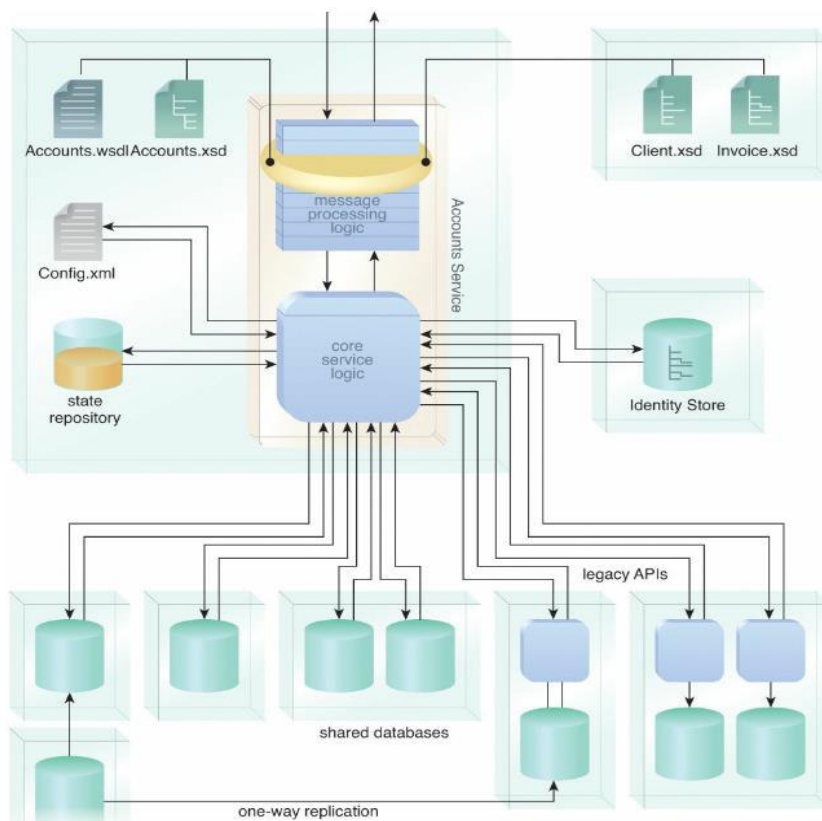
### ***Tipos de SOA***

Para logra un mejor entendimiento de las arquitecturas SOA debemos estudiar los tipos de SOA:

#### **Arquitectura de servicio**

De acuerdo a (Erl, 2016), una arquitectura de servicio se ocupa del diseño físico de un programa ofrecido como servicio. Esta arquitectura es parecida a la arquitectura de componentes con la diferencia a que esta puede abarcar mayor número de cosas como por ejemplo las extensiones de infraestructura que tendrá para suplir los requerimientos no funcionales como lo son más confiables, mejor rendimiento, escalabilidad, predictibilidad y en especial la de mayor autonomía.

En la figura siguiente se muestra un ejemplo de arquitectura de servicio:

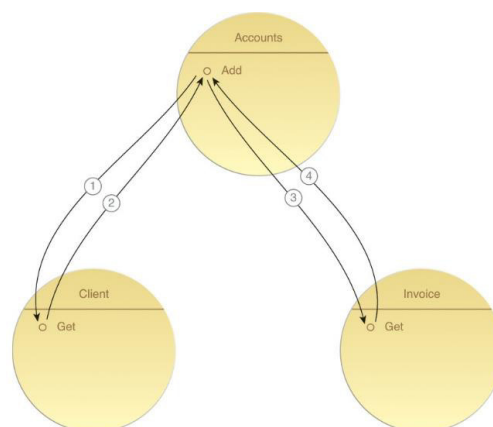


**Figura 2.16** Un ejemplo de una vista de arquitectura de servicios de alto nivel para el servicio Cuentas, que representa las partes de la infraestructura circundante utilizadas para cumplir con los requisitos funcionales de todas las capacidades. (Fuente: (Erl, 2016)).

### Arquitectura de composición de servicio

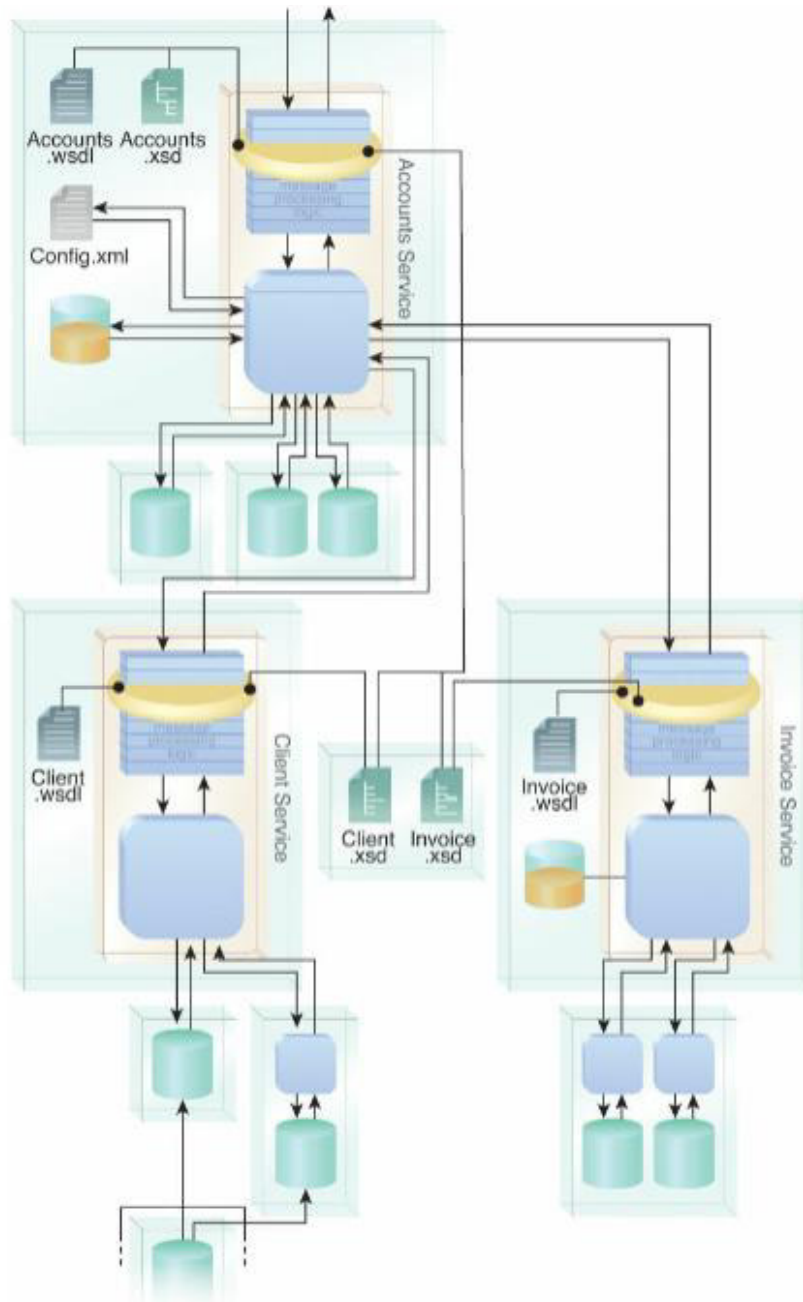
“El propósito fundamental de ofrecer una serie de servicios independientes es que se puedan combinar en composiciones de servicios, soluciones totalmente funcionales capaces de automatizar tareas del negocio más grandes y complejas” (Erl, 2016).

La siguiente figura muestra un ejemplo de composición de servicios, donde las flechas numeradas indican la secuencia del flujo de datos y la interacción del servicio requerida para la capacidad de “Add” para componer capacidades dentro de los servicios de Cliente y Factura (Erl, 2016):



**Figura 2.17** La composición del servicio de cuentas desde una perspectiva de modelado (Fuente: (Erl, 2016)).

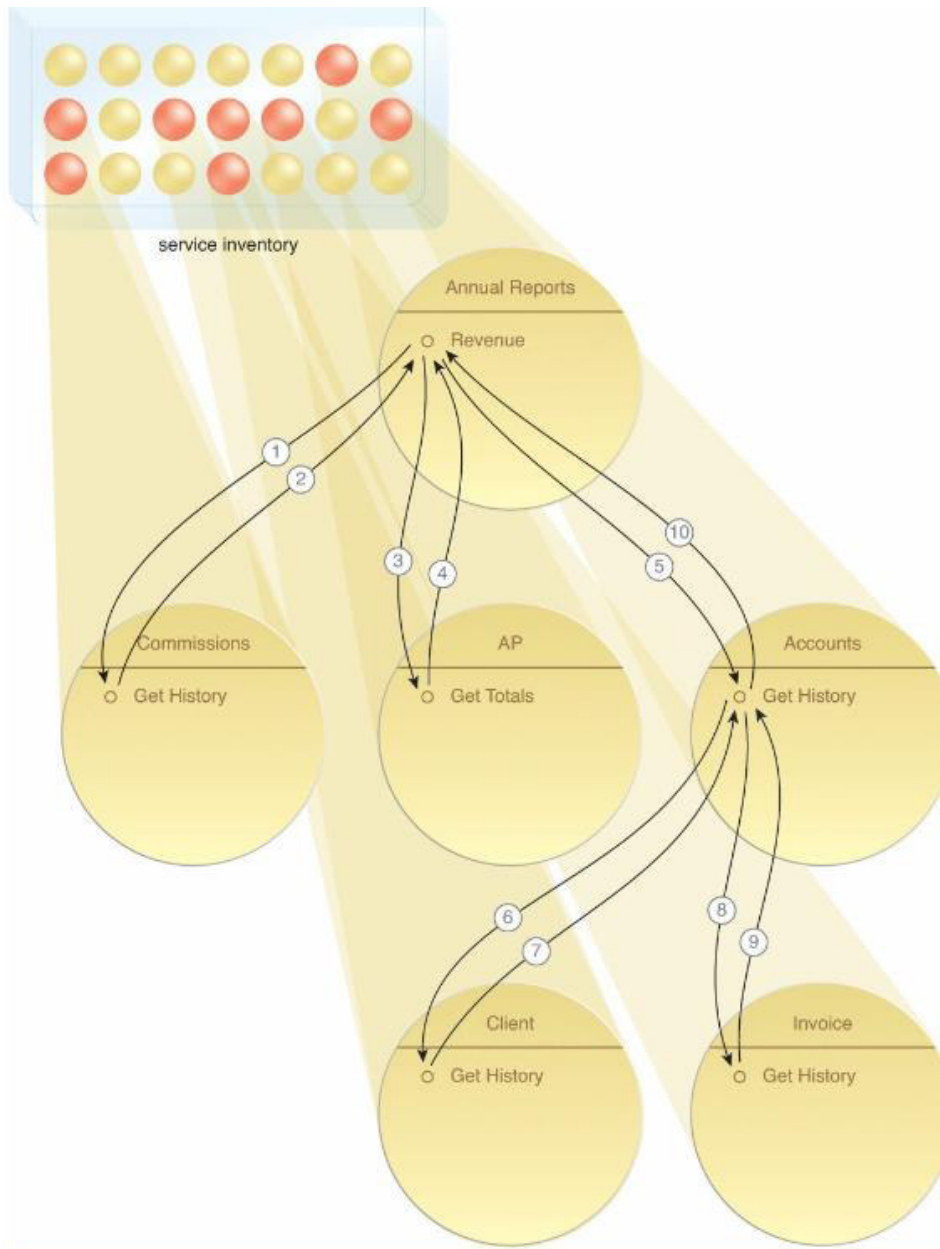
La misma composición de servicio de Cuentas de la figura anterior vista desde una perspectiva de arquitectura física que ilustra cómo los recursos subyacentes de cada miembro de composición proporcionan la funcionalidad requerida para automatizar la lógica de proceso representada por la capacidad de “Agregar” del servicio de Cuentas se muestra a continuación (Erl, 2016):



**Figura 2.18 Perspectiva de la arquitectura física de un servicio compuesto (Fuente: (Erl, 2016)).**

#### Arquitectura de inventario de servicio

Consiste en una colección de servicios normalizados y que son ofrecidos dentro de un marco predefinido. Generalmente abarca varios procesos del negocio y no solo uno (Erl, 2016).



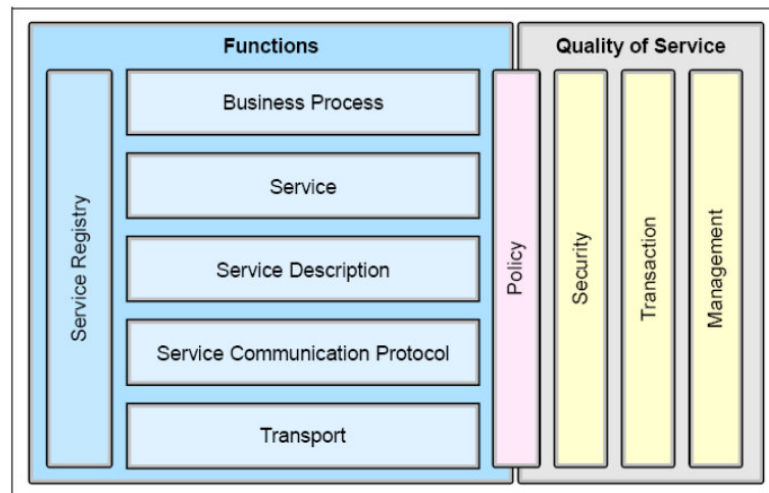
**Figura 2.19 Ejemplo de una arquitectura de servicio (Fuente: (Erl, 2016)).**

### **Arquitectura empresarial orientada a servicios**

“Esta forma de arquitectura tecnológica representa esencialmente todas las arquitecturas de servicios, composición de servicios e inventario de servicios que residen dentro de una empresa específica” (Erl, 2016).

### ***Componentes SOA***

A continuación, se muestra una pila arquitectónica y los elementos que se podrían observar en una arquitectura orientada a servicios:



**Figura 2.20 Elementos de una SOA (Fuente: (Endrei et al., 2004, p. 25))**

Según (Endrei et al., 2004, p. 25) la pila arquitectónica está dividida en dos mitades, la mitad izquierda aborda los aspectos funcionales de la arquitectura y la mitad derecha aborda los aspectos de la calidad del servicio. Estos elementos se describen en detalle de la siguiente manera:

#### **Aspectos funcionales**

Según M. Endrei, *et al.* los aspectos funcionales de una arquitectura SOA pueden incluir lo siguiente:

- El transporte es el mecanismo utilizado para trasladar las solicitudes de servicio del consumidor del servicio al proveedor del servicio y las respuestas del servicio del proveedor del servicio al consumidor del servicio.
- El protocolo de comunicación de servicio es un mecanismo acordado que el proveedor del servicio y el consumidor del servicio utilizan para comunicar lo que se solicita y lo que se devuelve.
- Descripción del servicio es un esquema acordado para describir qué es el servicio, cómo debe invocarse y qué datos se requieren para invocar el servicio con éxito.
- El servicio describe un servicio real que está disponible para su uso.
- Proceso del negocio es una colección de servicios, invocada en una secuencia particular con un conjunto particular de reglas, para cumplir con un requisito del negocio.
- El registro de servicio es un repositorio de servicios y descripciones de datos que puede ser utilizado por los proveedores de servicios para publicar sus servicios y servicios.

#### **Aspectos de calidad del servicio**

De acuerdo a (Endrei et al., 2004, p. 26) hay diversos aspectos de calidad del servicio de una SOA. Primero, la política, sirve para reglamentar la disponibilidad de los servicios a los



consumidores. Segundo, la seguridad, que permite reglamentar la identificación y el control de accesos para los consumidores del servicio. Tercero, la transacción, que son los datos que se aplican a un conglomerado de servicios para realizar una salida consistente, todos los servicios deben completarse si no ninguno se completa. Cuarto, la administración, que son los datos que se aplican a los servicios con la finalidad de ofrecer o consumirlos.

(Krafzig, Banke, & Slama, 2004), definen la estructura de SOA en cuatro abstracciones principales: servicio, repositorio de servicios y bus de servicios en la siguiente figura:



**Figura 2.21 Diagrama de componentes de una arquitectura SOA (Fuente: (Krafzig et al., 2004))**

Como se apreció antes dos conceptos muy importantes para entender la arquitectura SOA son el concepto de servicio y el concepto de web services (D. Barry y D.Dick). Según explican Krafzig, Banke y Slama, es una tarea automatizada que expone datos y lógica del negocio y que está compuesta de contratos de servicios que especifican la funcionalidad, interfaces por la que expone la funcionalidad de cada servicio, el repositorio en el cual guarda los contratos de los servicios y el bus que comunica a los servicios y las interfaces de la aplicación (Krafzig et al., 2004).

En este capítulo se expuso teoría bibliográfica referente a la gestión de ventas, al proceso de ventas, los sistemas de información, a las aplicaciones móviles y sus tipos, a las aplicaciones distribuidas y a los estilos arquitectónicos candidatos para la solución que se quiere plantear. En el siguiente capítulo se expondrá los puntos referentes al arte metodológico para la investigación que comprende temas como los trabajos relacionados y algunas soluciones existentes en el mercado parecidas a la solución propuesta.

## **CAPÍTULO III. ESTADO DEL ARTE METODOLÓGICO**

En el presente capítulo se describen los conocimientos existentes en la literatura relacionados a la implementación de la aplicación a implementar. Inicialmente se empieza por describir los trabajos relacionados para esta investigación, tratando aquí el problema, justificación, solución tecnológica (y el estilo arquitectónico que se utiliza para su construcción) y los beneficios de la solución seleccionada que tiene cada uno. Finalmente, se brinda una descripción de los aplicativos para ventas en el mercado que son más semejantes a lo que se quiere desarrollar.

### **3.1 Trabajos relacionados**

#### **3.1.1 Diseño e implementación de una aplicación distribuida de gestión de inventario para dispositivos móviles.**

Según lo descrito por (Víctor, 2011), este trabajo se fundamenta en el análisis, el diseño e implementación de una aplicación distribuida para la gestión de inventarios que pone a disposición del sector empresarial.

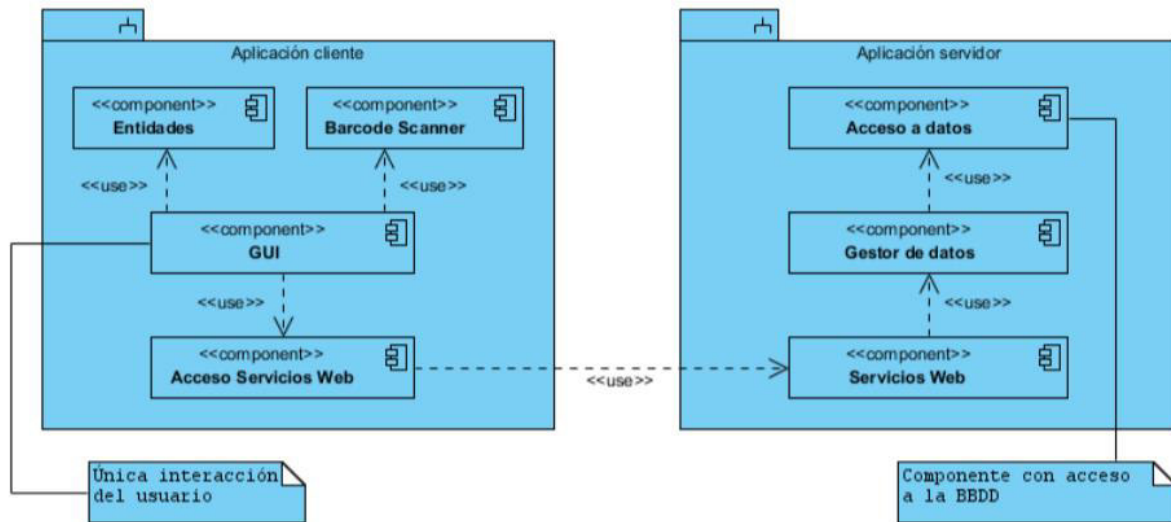
El problema que se aborda en este proyecto de investigación, según (Víctor, 2011), es la forma poco eficiente, poco segura y engorrosa de la gestión de inventarios para los usuarios de las pymes, grandes empresas e instituciones públicas.

Este proyecto se justifica por la gran demanda de hacer más eficiente, segura y sencilla la gestión de inventarios en las compañías. La necesidad de realizar este proceso más eficiente, motivada por el incremento en la capacidad y velocidad de transmisión de datos en la red, la crecida en el uso de los móviles inteligentes tanto personal como empresarialmente, la mayor facultad de procesamiento y de almacenamiento de información en los móviles (Víctor, 2011).

La solución tecnológica implementada por (Víctor, 2011) consiste de dos componentes: una aplicación móvil basada en Android y una aplicación del lado del servidor que consiste en web services que acceden a los datos de una base de datos centralizada y que contienen el dominio de la gestión de inventarios. Utiliza el estilo arquitectónico de software cliente-servidor en conjunción con el patrón MVC (Modelo-Vista-Controlador). En la aplicación del patrón arquitectónico de este sistema, el controlador se distribuye tanto en el cliente como en el servidor. En el servidor se gestiona la data del modelo y también la comunicación con el cliente a través de web services. En el dispositivo móvil, se realiza la lógica de la navegación de

páginas, la gestión de los errores, la serialización y deserialización de datos en JSON, el envío de peticiones y recepción de mensajes del servidor.

En la figura siguiente se muestra diagrama de componentes del sistema desarrollado en este trabajo:



**Figura 3.1 Diagrama de componentes utilizado por la aplicación distribuida de gestión de inventario para dispositivos móviles (Fuente: (Víctor, 2011))**

El beneficio de esta solución es la gestión de inventario más rápida, fiable, simple y fácil. Permite el acceso remoto a la información originando ahorro en tiempo, dinero en las actividades de actualización de datos, permitiendo además que los trabajadores puedan realizar su trabajo desde cualquier parte lo que brinda una mejor flexibilidad y rapidez para este proceso del negocio (Víctor, 2011).

### **3.1.2 Desarrollo de aplicaciones distribuidas online y smart client para dispositivos móviles utilizando .NET, que permitan acceder al sistema de administración estudiantil.**

De acuerdo a (Armijos & Pelaéz, 2008), este trabajo tiene por objetivo el desarrollo de dos aplicaciones (una aplicación online y una aplicación “smart client”) y de un sistema proveedor de servicios para éstas mediante servicios web.

El problema que pretende abordar según se infiere de (Armijos & Pelaéz, 2008), es que el proceso de consulta de horarios, de materias y de calificaciones de los alumnos de las carreras de Electrónica y Redes de Información de una universidad es poco eficiente.

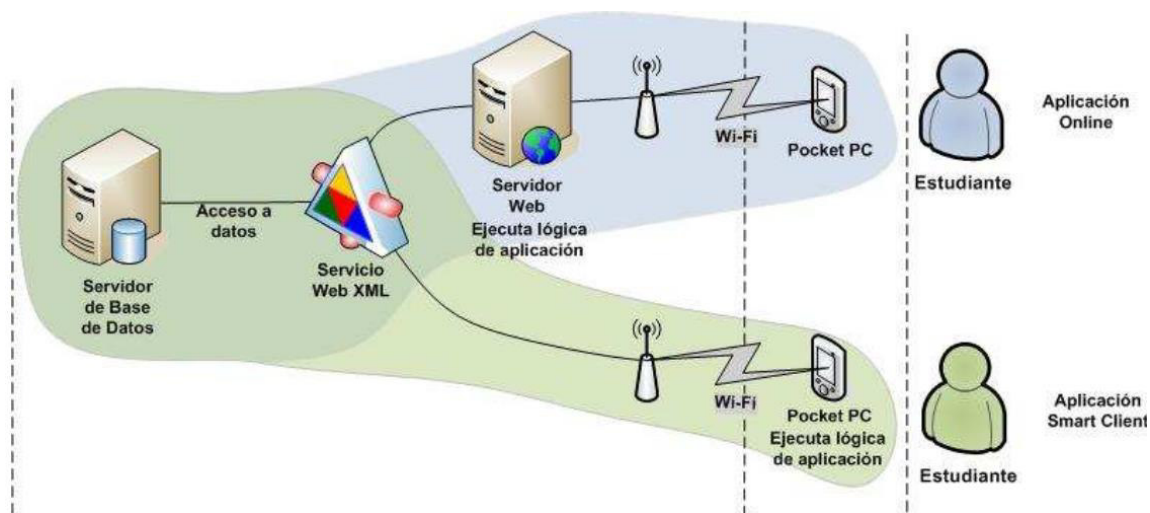
Este proyecto se justifica por la necesidad de tener una alternativa más eficiente para la consulta de las calificaciones, cursos y horarios de los periodos. La implementación de estas aplicaciones

para cubrir dichas necesidades se justifica porque gran sector de los alumnos de dichas facultades, tienen dispositivos móviles como teléfonos móviles y dispositivos móviles Pocket PC (PDAs), que pueden acceder a Internet y que pueden ser utilizados para realizar las consultas de forma más rápida (Armijos & Pelaéz, 2008).

(Armijos & Pelaéz, 2008) nos explica su solución tecnológica implementada en los siguientes pasos: Primero, la construcción de los web services en dos capas: una de acceso a datos que ejecuta procedimientos almacenados y otra de lógica del negocio que corresponde a un conjunto de clases que encapsulan los datos obtenidos del SAE. Segundo, la aplicación online que consume y muestra los datos obtenidos del web service, para ello se usa ASP.NET Mobile e Internet Information Server (IIS). Tercero, una aplicación Smart Client que consume el servicio web, utilizando Compact Framework para Pocket PC.

El estilo arquitectónico de software que se utiliza es el de cliente servidor, donde el componente proveedor de servicios hace las veces de servidor y las dos aplicaciones las veces de clientes.

En la siguiente figura se muestra el despliegue de la solución implementada en este trabajo:



**Figura 3.2 Diagrama de despliegue de la aplicación Web services, Aplicación Online y aplicación Smart Client (Fuente: Adaptado de (Armijos & Pelaéz, 2008))**

Los beneficios de esta solución tecnológica son el contar con un canal más para poder realizar las consultas por parte de los alumnos. La ventaja de poder realizar dichas consultas sin necesidad de tener que asistir a la oficina para poder recibir la información por parte del personal encargado. La ventaja de tener menos carga de trabajo por parte del personal encargado, sin necesidad de estar atendiendo a todos los alumnos debido a que muchos de ellos podrán usar este nuevo canal de información.

### **3.1.3 Desarrollo de un sistema de Información de Registro de pedidos para Ventas usando dispositivos móviles**

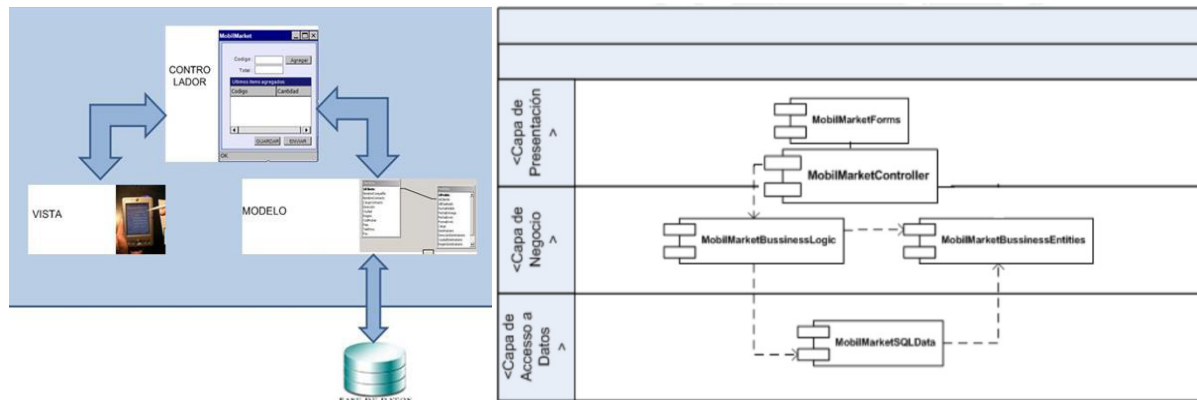
En este proyecto se desarrolla un sistema de información de registro de pedidos para móviles para compañías medianas. De tal forma que esta actividad se realice a través de Internet. Que provea la información de los clientes, productos de forma eficiente además de reportes que ayuden a tomar decisiones, por ejemplo, se construye la funcionalidad de reportes de consolidado de pedidos por vendedor, o reporte de productos más vendidos (Balarezo, 2012).

El problema que aborda este trabajo de acuerdo a (Balarezo, 2012) es que la captación de clientes, y de pedidos por los representantes de ventas de gran parte de las empresas es de forma manual en formularios impresos. Esta información al terminar el día se ingresa a los sistemas para procesar el pedido. Lo cual genera malestar debido a la urgencia de los clientes para que sus pedidos sean atendidos pronto. Cabe la posibilidad de que a veces se pierda la información debido al extravío de los papeles lo que ocasionaría que el pedido no sea atendido y que por consiguiente el cliente piense en proveedores o productos alternativos.

(Balarezo, 2012), justifica este trabajo gracias a la necesidad de recuperar la información de los productos y de registrar los pedidos desde donde se encuentren los vendedores, por ejemplo, cuando estos salen al campo a vender sus productos o a realizar estudios de mercado. Además, de que la solución no sea costosa debido a que en el Perú la mayoría de compañías no cuentan con mucho dinero para adquirir una solución del mercado.

La solución tecnológica sigue el estilo arquitectónico de software cliente-Servidor en conjunción con el patrón de arquitectura MVC. El lado del cliente se ejecuta en los móviles de los vendedores, el cual es implementado con la plataforma .NET con Visual Basic. El lado del servidor se ejecuta en los computadores de la compañía conformados por servicios web y procedimientos almacenados en Sql Server para la recuperación y persistencia de los datos (Balarezo, 2012).

En la figura siguiente se muestra el patrón de diseño aplicado al proyecto y la vista de implementación para la arquitectura utilizada para este trabajo respectivamente:



**Figura 3.3 Patrón de diseño MVC y vista de implementación de la arquitectura del sistema Información de Registro de pedidos de Ventas usando dispositivos móviles**  
(Fuente: (Balarezo, 2012))

Los beneficios de esta solución son la disminución en el tiempo para comenzar a procesar el pedido y evitar la pérdida de información. La eliminación de los formularios impresos. La solución ofrecida tiene un costo que aproximadamente equivale a la quinta parte del precio de otras soluciones en el mercado haciéndola más asequible para los bolsillos de las compañías peruanas.

## 3.2 Aplicaciones software

A continuación, se detallan sistemas que cuentan con soluciones afines a la planteada:

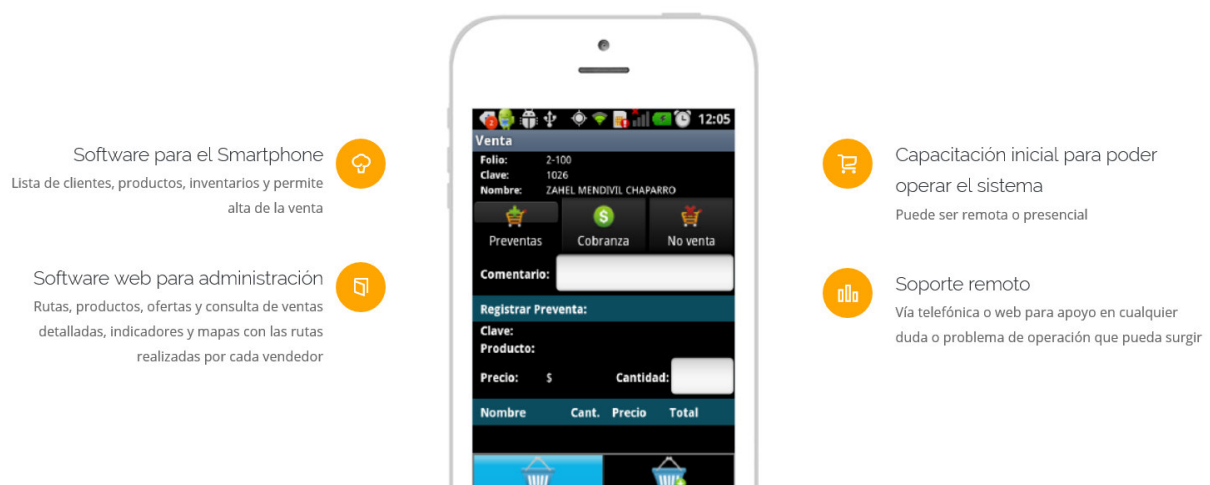
### 3.2.1 Vemobile Ventas Express Mobile

De acuerdo a Vemobile, ésta es una solución tecnológica que permite a los vendedores realizar sus tareas en ruta de forma más precisa, rápida de tal forma, que mejora la atención y la calidad del servicio al cliente, este producto está disponible para móviles Android o Windows y está compuesto de 2 componentes (Vemobile, 2014).

El primero, es un aplicativo móvil que tiene como usuarios a los vendedores y los supervisores de ruta, también cuenta con una agenda de visita para su operación, y desde este componente, también se pueden realizar acciones como: “[...] revisar toda la información general del cliente, el inventario de productos disponibles, elegir precios desde una o varias listas, ver las características de los productos, levantar el pedido, acumular partidas, validar el total de la venta” (Vemobile, 2014). Además (Vemobile, 2014), especifica que se pueden realizar acciones como el envío del pedido a la oficina para su atención y la impresión de tickets para ventas a través de una impresora móvil.

El segundo componente es una Aplicación WEB que “[...] organiza las rutas, administra la agenda de los vendedores, actualiza los catálogos de clientes, productos y precios. En ésta se establecen los criterios y políticas de ventas, descuentos y promociones” (Vemobile, 2014). Además, permite “[...] visualizar gráficamente en cualquier momento el desempeño puntual de cada ruta/vendedor. Se pueden monitorear los indicadores de ventas como clientes visitados, visitas con o sin venta, causas de no venta, productos más vendidos, comparativo de ventas vs. metas y más.” (Vemobile, 2014).

A continuación, se muestra un formulario de la interfaz gráfica de usuario de la aplicación presentado en el sitio web de la aplicación:



**Figura 3.4 Interfaz gráfica de usuario de una de las funcionalidades de Ve Mobile en su portal web (Vemobile, 2014).**

De acuerdo a Vemobile, ésta es una herramienta para empresas de delivery pequeñas y también para empresas medianas y grandes que cuenten con un sistema ERP.

### 3.2.2 Sevenminds

#### Solución de software como servicio

“Sevenminds es una solución de software como servicio (SaaS). Solución para captura de datos con PC’s o dispositivos móviles, desarrollada para compañías de múltiples sectores que requieren obtener información de campo desde diferentes lugares” (Sevenminds, 2013c).

Según (Sevenminds, 2013b) , esta solución habilita la recolección de datos de visitas a clientes o de puntos de venta, tomas de pedidos o datos del producto en tiempo real, consolidación de información y generación de portales web.

#### Funcionamiento

(Sevenminds, 2013c), describe el funcionamiento de su solución ofrecida:

Cada ejecutivo de ventas va equipado con un software para dispositivos móviles (Laptops, Tablets, Smartphones, celulares, etc.) que tiene cargado su plan de trabajo diario incluyendo sus objetivos y su ruta. A medida que el personal en campo realiza las visitas a sus clientes va enviando datos en tiempo real a los servidores de Sevenminds.

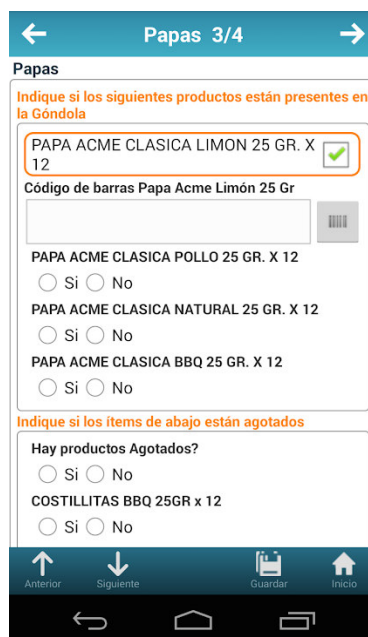
A través de la nube la dirección y toda la estructura comercial pueden hacer seguimiento de la ejecución de la estrategia de ventas a través de informes e indicadores de gestión en línea, que permiten conocer los resultados comerciales y tomar decisiones a partir de información actualizada. (Sevenminds, 2016).

### Arquitectura y tecnología

(Sevenminds, 2013a) nos menciona que su solución SAS permite reducir costos debido a que para utilizarlo no hay necesidad de montar una infraestructura completa debido a que el sistema se encuentra operando en los servidores del cliente, por lo que no se necesita gastos en compra de hardware adicional. Proporciona un sistema web que se puede ingresar y operar de manera remota. También, nos presenta las tecnologías de desarrollo de este SAS: “[...]Asp.Net Framework 4.0 – Microsoft, SQL Server 2008 Enterprise – Microsoft., IOS 5 – Apple, Android 4.0 – Google., Suite Gráfica - Component Art., Firewall – Cisco Systems, Servidores – Dell, SSL” (Sevenminds, 2013a).

Esta solución está optimizada para plataformas móviles de Android y iPhone.

La siguiente figura muestra una interfaz gráfica de usuario de esta aplicación:



**Figura 3.5 Interfaz de usuario Sevenminds (Fuente: (Sevenminds, 2016))**

### 3.2.3 Administración de Ventas Móviles (Field Service)

(Bpmcapital, 2013) nos describe su producto:



La Administración de Ventas Móviles (Field Service) de BPM capital proporcionan una visión instantánea de los datos e información que residen en la oficina, permitiendo una mejor organización para la gestión de toda la cadena de servicios de campo desde el cliente, representante de ventas, técnico y de servicio al proveedor, y más allá. Nuestras aplicaciones permiten ayudar a entregar información en el punto de venta a su equipo de ventas cuando más lo necesitan. Al mismo tiempo que permite a representantes de ventas a cerrar más negocios mientras están en el campo (calle) con los clientes.

En la figura siguiente se muestra una interfaz de usuario de la aplicación:



**Figura 3.6 Administración de Ventas Móviles -Field Service (Fuente: (Bpmcapital, 2013))**

Según (Bpmcapital, 2013), las ventajas de este producto son listadas a continuación:

#### **Visibilidad del inventario**

Aumente la precisión de los datos y proporcione mejores capacidades de administración del inventario a través del tiempo real o la visibilidad o por lotes parciales del nivel de visibilidad.

#### **Gestión de Ventas**

Proporcionar una visión instantánea al gerente de administración de ventas para facilitar los procesos de toma de decisiones de ventas en el campo.

#### **Satisfacción del Cliente**

Mejorar su capacidad para entregar los pedidos especiales, impulsar los ingresos incrementales, y proporcionar información precisa sobre inventario, envío y plazos de entrega on-the-fly.

### **Ventas locales**

Mejora de los niveles de servicio para los clientes, proporcionando información instantánea de la cuenta de un cliente, estado del pedido, la disponibilidad de inventario y los plazos de entrega para ayudar a las empresas más cercanas.

### **Información total al cliente**

Los datos completos del cliente, información de cuentas, estado de los pedidos, la historia de ventas, los precios de los productos precisos, promociones, inventario y manejo de cobro.

En este capítulo se expuso los puntos referentes al arte metodológico que comprende temas como los trabajos relacionados y algunas soluciones existentes en el mercado parecidas a la solución propuesta

En el siguiente capítulo se expondrá lo referente al aporte teórico donde se abordará el proceso de selección del estilo arquitectónico de software y a continuación la adaptación de dicho estilo a la solución propuesta.

## **CAPÍTULO IV. APOORTE TEÓRICO**

Dentro de este capítulo, el apartado de selección y justificación del estilo arquitectónico consiste en la evaluación comparativa entre los estilos arquitectónicos candidatos. Primeramente, se empieza por listar las arquitecturas candidatas para la implementación de la aplicación, luego se definen y justifican los criterios de evaluación a tomar en cuenta para dicha selección, seguidamente, se realiza el análisis comparativo de dichos estilos arquitectónicos tomando en cuenta el contexto en el cual se está desarrollando este trabajo.

En el apartado de adaptación de la arquitectura de software se representa y se describe el modelo de la solución planteada en un alto nivel de abstracción mediante una ilustración donde se adapta el estilo arquitectónico a la aplicación a desarrollar.

### **4.1 Selección y justificación del estilo arquitectónico**

#### **4.1.1 Listado de posibles arquitecturas para la aplicación.**

El arquitecto de software debe comparar y contrastar las alternativas disponibles con el fin de sopesar la mejor solución. Comparar las arquitecturas de software para cualquier sistema no trivial es una tarea difícil, no importa cómo muchas organizaciones requieren arquitecturas candidatas a largo plazo. A continuación, se listan las 3 posibles arquitecturas para la aplicación en cuestión y que seguidamente será seleccionada una de ellas tomando en cuenta criterios de selección adecuados:

- Arquitectura cliente servidor.
- Arquitectura de n capas.
- Arquitectura orientada a servicios.

#### **4.1.2 Definición y Justificación de los Criterios de Evaluación**

El principal objetivo de este análisis es encontrar la respuesta a la pregunta: ¿Qué estilo arquitectónico ofrece o promueve la mejor realización de los requerimientos no funcionales que son necesarios satisfacer para esta aplicación necesarios para la implementación de este sistema?

Los criterios de evaluación que se tomarán en cuenta para elegir el estilo arquitectónico adecuado parten de los requerimientos no funcionales para la solución planteada y son:

1. **Costo:** Representa el nivel de costo de la implementación de todas las funcionalidades del sistema.

2. **Dificultad:** Representará el nivel de dificultad de implementación del sistema de un determinado estilo arquitectónico.
3. **Mantenibilidad:** Mide la facilidad con que puede darse mantenimiento al software.  
De acuerdo a (Visser, 2016, p. 2) existen 4 tipos de mantenimiento de software:  
Los errores que son descubiertos y tienen que ser corregidos (mantenimiento correctivo). El sistema tiene que ser adaptado a cambios en el entorno en el que opera, por ejemplo, cuando se hace actualizaciones del sistema operativo o de tecnologías (mantenimiento adaptativo). Los usuarios del sistema (u otros stakeholders) tienen requisitos nuevos o modificados (mantenimiento perfectivo). Se identifican las formas de aumentar la calidad o evitar errores futuros (mantenimiento preventivo).
4. **Escalabilidad:** Éste especificará el tamaño de crecimiento con que el producto manejará, la expectativa de crecimiento. Según (Bondi, 2015), el concepto connota la capacidad del sistema para incorporar elementos u objetos, para procesar volúmenes de trabajo crecientes.
5. **Interoperabilidad:** La interoperabilidad en las aplicaciones empresariales se puede definir como la capacidad de un sistema o producto para trabajar con otros sistemas o productos sin un esfuerzo especial del cliente o usuario (Konstantas, Bourrières, Léonard, & Boudjlida, 2005).
6. **Reusabilidad:** Según el Instituto de Ingenieros Eléctricos y Electrónicos, la reusabilidad es el grado en que un módulo de software u otro producto de trabajo se puede usar en más de un programa de computadora o sistema de software (IEEE, 1991, p. 174).

#### 4.1.3 Análisis comparativo de las arquitecturas de software

##### ❖ Arquitectura cliente servidor

###### Ventajas:

**Escalabilidad:** Se pueden incrementar el número de clientes asociado a un determinado servidor. Según (Joines, Willenborg, & Hygh, 2002), se puede escalar tanto de forma vertical como horizontal por ejemplo al añadir CPU o memoria para el servidor se puede lograr mayor rendimiento para la carga de trabajo originada desde los clientes. Según (Saternos, 2014, p. 41), existen diseños sin estado en la arquitectura cliente servidor (REST) que permiten que toda la petición del cliente incluya el estado completo del mismo, así que el servidor no se encargará

del manejo de la sesión de los clientes. De tal modo que servidores pueden ser fácilmente añadidos para suplir el incremento de carga para facilitar la escalabilidad horizontal.

**Dificultad:** “Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo” (Ferrer, 2012).

**Interoperabilidad:** Distribución de aplicaciones: concurrencia de procesos. Permite emplear entornos heterogéneos: la plataforma no tiene por qué ser la misma (se comunican a través de un API conocido) (Konstantas et al., 2005).

**Mantenibilidad:** Esta arquitectura es de fácil mantenimiento debido a que: “[..] al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente)” (Ferrer, 2012, p. 22). Además, de acuerdo a (Maffeis & Olsen & Associates, 1997), este modelo es una extensión del modelo de programación modular basada en objetos donde las piezas largas de software son estructuradas en componentes más pequeños que tienen interfaces bien definidas. Este modelo crea la ventaja de una mejor mantenibilidad. Además, es importante considerar la división de la lógica del cliente y del servidor ya que tiene un impacto importante en la mantenibilidad en esta arquitectura.

**Costo:** Utilizar la arquitectura cliente/servidor es económico, comparado con la amplitud y el alcance y tiempo de implementación de una arquitectura SOA, ésta resulta ser más económica. La ventaja de utilizar esta arquitectura es que cada vez hay más tecnologías (hardware y software) tanto del lado cliente y del lado servidor con mayores capacidades, a menor costo y de distintos proveedores lo cual contribuye considerablemente a la reducción de costos.

#### **Desventajas:**

**Reusabilidad:** Cuando se implementa una arquitectura cliente / servidor de 3 capas, en comparación con la arquitectura de 2 capas, incrementar el rendimiento, flexibilidad, mantenibilidad, reusabilidad y escalabilidad mientras se esconde la complejidad del procesamiento distribuido al usuario (González, 2000). Sin embargo, cuando se utiliza cliente servidor de n capas se obtienen los mismos beneficios, pero conlleva mayor dificultad de implementación.

#### **❖ Arquitectura en n capas**

## **Ventajas:**

**Mantenibilidad:** Permite modificar una capa sin tener que modificar toda la aplicación gracias a la “separación de responsabilidades entre componentes (por ejemplo, separar el interfaz de usuario de la lógica de negocio, y la lógica de negocio del acceso a la base de datos) aumenta la flexibilidad, mantenibilidad y escalabilidad” (De la Torre et al., 2010, p. 35).

**Reusabilidad:** Se puede usar cuando: “Ya tienes construidas capas de una aplicación anterior, que pueden reutilizarse o integrarse. Ya tienes aplicaciones que exponen su lógica de negocio a través de interfaces de servicios” (De la Torre et al., 2010, p. 16). También se puede construir una capa de servicios distribuidos (web services) en la aplicación para brindar servicios a otras aplicaciones, y así facilitar la reusabilidad del modelo y de la lógica de las aplicaciones.

**Escalabilidad:** En esta arquitectura, “[...] distribuyendo las capas en distintos niveles físicos se puede mejorar la escalabilidad, la tolerancia a fallos y el rendimiento” (De la Torre et al., 2010, p. 15).

**Interoperabilidad:** Dependiendo del alcance del sistema, se puede implementar una capa que del sistema que ofrezca un api de comunicación y así facilitar la interoperabilidad con otros sistemas (por ejemplo, la implementación de servicios web).

**Costo:** El costo variará de acuerdo a la complejidad del sistema. Considerando además que se pueden utilizar mayor número de capas conforme éste sea más complejo. Requisitos como la alta disponibilidad, seguridad, tecnologías usadas pueden incrementar el número de capas que se utilicen y así también el costo del sistema.

**Dificultad.** A medida de la complejidad del sistema se puede incrementar el número de capas y/o puede incluir mayor número de tecnologías de tal forma que puede incrementarse la dificultad de implementación posiblemente.

## **❖ Arquitectura orientada a servicios**

### **Ventajas:**

**Interoperabilidad.** Los programas de software más interoperables son los que más fácil resulta intercambiar información. Programas de software que no son interoperables necesitan ser integrados. Programas de software que no son interoperable necesitan ser integrados. Por lo tanto, la integración puede ser visto como un proceso que habilita la interoperabilidad. Un objetivo de la orientación a servicios es establecer interoperabilidad nativa dentro de los servicios para reducir la necesidad de integración. (Erl, 2016).

**Reusabilidad:**

De acuerdo a (Barry & Dick, 2013), la reusabilidad es inherente a este estilo arquitectónico: “Un servicio es software y hardware. Uno o más servicios soportan o automatizan una función del negocio. Mas a menudo, el intento es que un servicio pueda ser usado en múltiples formas (frecuentemente referido como usabilidad)”.

Según (Hau, Ebert, Hochstein, & Brenner, 2008), SOA es supuesta para proveer servicios que puedan ser usados en más de un contexto. Por ejemplo, en un escenario multicanal. La misma data tiene que ser presentado a usuarios en diferentes dispositivos o data que tiene que ser enviada de diferentes dispositivos a un receptor. En vez de programar todas las tuberías a y desde cada dispositivo, la aplicación debería proveer servicio único que agrupe la funcionalidad necesaria. Esta solución sería mucho más eficiente que el enfoque de “hacerlo n veces”.

**Escalabilidad:** Según (Rotem-Gal-Oz, 2012), existen patrones de arquitectura orientada a servicios (SOA) para incrementar atributos de calidad como la escalabilidad. Por ejemplo, se tienen los patrones: 1) Patrón de tuberías paralelas que incrementa el paralelismo de la aplicación cuando se añade cada tubería y el rendimiento se incrementa como resultado. Este patrón incrementa la escalabilidad del servicio tremendamente. 2) El patrón de componentes de infraestructura de red (el agente, nodo raíz, etc.) constantemente monitorean la disponibilidad de recursos. Al añadir hardware configurado con los componentes de la red agranda el pool de recursos disponibles. La red se encarga de maximizar el uso y lo hace basándose en la carga de las máquinas. Esta asignación de recursos “inteligente” ayuda a resolver la escalabilidad y el balanceo de carga. Adicionalmente, en la red implementa redundancia y tolerancia a fallos y puede pasar tareas a nuevos nodos cuando un nodo falla. 3) Patrón de instancia de servicio consiste en el despliegue de copias múltiples de los servicios usando un despachador en el borde que distribuirá el trabajo a las diferentes instancias. El punto importante es que los consumidores del servicio no serán consientes y no afectados por el escalamiento que ocurre dentro del servicio y es más adecuado para implementaciones de servicios sin estado.

**Mantenibilidad:** (Rotem-Gal-Oz, 2012) recalca que, por definición, SOA trae beneficios de arquitectura a un Sistema de software distribuido. Muchos atributos de calidad son abordados (reusabilidad, adaptabilidad, mantenibilidad) entre ellos la mantenibilidad en el cual los servicios pueden ser mantenidos por equipos dedicados, más pequeños, y pueden ser probados de esta forma también. Estos beneficios existen porque SOA elimina los problemas relacionado a la integración punto a punto.

### Desventajas:

**Costo:** (Gerencia, 2014) nos habla que las empresas tienen la misma problemática independientemente de su envergadura y que cualquier empresa, independientemente de su tamaño, puede aprovechar las ventajas de SOA; sin embargo, la implantación de SOA requiere de recursos considerables. De lo que podemos deducir que esta arquitectura demanda mayor costo que la implementación de otras arquitecturas.

**Dificultad:** De acuerdo a (Rotem-Gal-Oz, 2012) hay patrones de arquitectura orientada a servicios que pueden resolver la mayoría de las necesidades de los atributos de calidad de software. Pero el uso de estas tecnologías puede ser relativamente complicadas y costosas.

Una vez mencionado las ventajas y desventajas de los estilos arquitectónicos mencionados, se realizará la evaluación comparativa tomando como base los criterios de comparación descritos anteriormente y centrándonos en la resolución del problema, la cual busca implementar la aplicación antes mencionada.

En primer lugar, se asigna a cada uno de los criterios de evaluación, el peso correspondiente en base a los requisitos no funcionales, como se muestra en la siguiente tabla:

	Costo	Dificultad	Reusabilidad	Escalabilidad	Mantenibilidad	Interoperabilidad
Peso	35%	20%	20%	10%	10%	5%

**Tabla 4.1 Pesos para los Criterios de Evaluación (Fuente: Elaboración propia)**

Así mismo, en la tabla que sigue, se indica los valores de las notas (cuadro amarillo) y el valor de las notas ponderadas (cuadro gris) que se obtuvieron al multiplicar el peso por cada nota correspondiente a cada uno de los criterios de evaluación, y clasificándolas con los ítems alto, medio y bajo considerando que la máxima nota a obtener es 20 y la mínima 0.

<u>Nota</u>	Costo		Dificultad		Reusabilidad		Escalabilidad		Mantenibilidad		Interoperabilidad	
Alto	0	0	0	0	20	4	20	2	20	2	20	1
Medio	10	3.5	10	2	10	2	10	1	10	1	10	0.5
Bajo	20	7	20	4	0	0	0	0	0	0	0	0

**Tabla 4.2 Notas para los Criterios de Evaluación (Fuente: Elaboración propia)**



Una vez mencionadas las ventajas y desventajas de los métodos de pronóstico, y luego de haber establecido los pesos y las notas para cada criterio de evaluación. Se procede a elaborar el cuadro comparativo de las arquitecturas de software, obteniendo la nota final correspondiente a cada uno de ellos, como se muestra la tabla siguiente:

Arquitectura	Criterio de Evaluación					
	Costo	Dificultad	Reusabilidad	Escalabilidad	Mantenibilidad	Interoperabilidad
Arquitectura cliente servidor	Bajo	Medio	Medio	Medio	Alto	Medio
Arquitectura de n capas	Medio	Medio	Alto	Alto	Medio	Medio
Arquitectura orientada a servicios	Alto	Alto	Alto	Alto	Alto	Alto

**Tabla 4.3 Cuadro de Evaluación Comparativo de Arquitecturas de software (Fuente: Elaboración propia)**

Finalmente, en la siguiente tabla se muestran las notas correspondientes a cada arquitectura evaluada obtenida mediante el proceso de comparación.

Arquitectura	Arquitectura cliente servidor	Arquitectura de n capas	Arquitectura orientada a servicios
Nota	7+2+2+1+2+0.5=14,5	3,5+2+4+2+1+0,5=13	0+0+4+2+2+1=9

**Tabla 4.4 Notas ponderadas correspondientes a cada arquitectura de software (Fuente: Elaboración propia)**

#### 4.1.4 Arquitectura Seleccionada

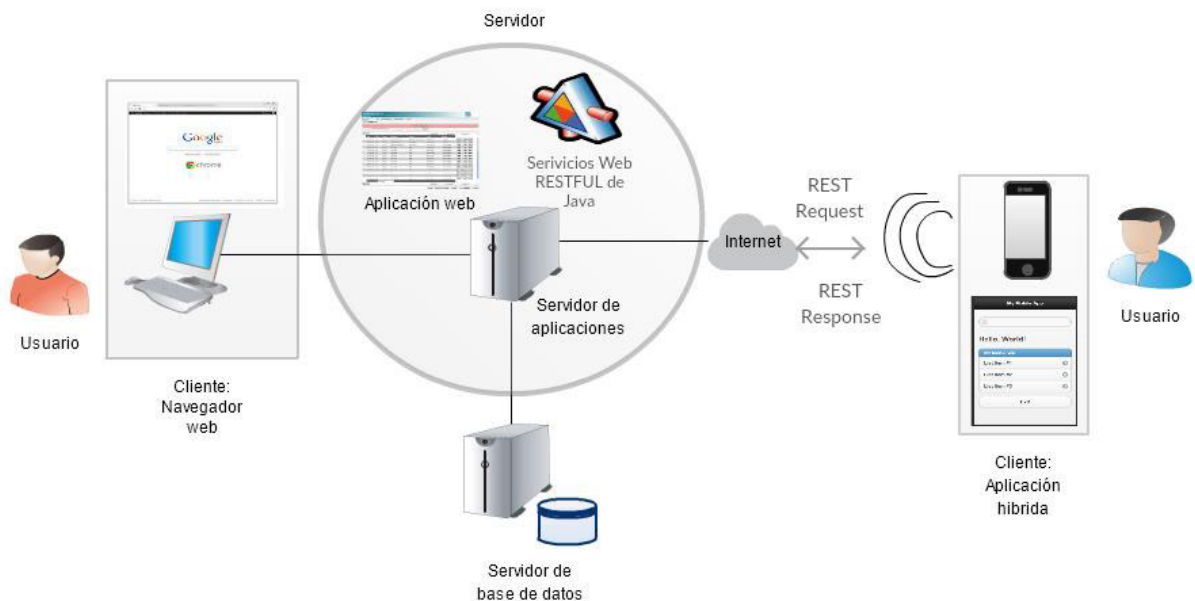
Para la implementación del sistema se opta por seleccionar como arquitectura global del sistema al estilo arquitectónico cliente servidor, ya que cuenta con la mejor nota en el análisis anterior.

Observando las necesidades de comunicación para la aplicación distribuida, el componente cliente se comunica con el componente servidor para el intercambio de datos. Para ello la opción que elijo para la implantación de esta comunicación es la comunicación a través de web services RESTful. Por lo que para ser más específicos de acuerdo a (Saternos, 2014, p. 6) la arquitectura general del sistema que se está eligiendo es la Arquitectura Web Cliente servidor.

Debo resaltar que el resultado obtenido se debe a que es más asequible en términos económicos y de mayor facilidad de implementación ya que es una estructura más simple, y que compromete un menor número de elementos en su estructura en comparación por ejemplo a la de una arquitectura orientada a servicios.

## 4.2 Adaptación de la arquitectura de software

La arquitectura web cliente servidor elegida anteriormente se tiene que adaptar para la solución que se quiere implementar, para eso, la aplicación distribuida que es una aplicación con distintos componentes que se ejecutan en entornos separados, normalmente en diferentes plataformas conectadas a través de una red. En la figura siguiente se muestra la arquitectura seleccionada y la adaptación que se realizará para el sistema a desarrollar.



**Figura 4.1 Arquitectura del sistema a implementar (Fuente: Elaboración propia)**

La aplicación distribuida mostrada en la figura anterior sigue el modelo cliente-servidor y tiene los siguientes componentes:

**Lado del servidor:** Estará conformado por un servidor de aplicaciones que hará uso del servidor de base de datos que permitirá la inserción y recuperación y de los datos en una base de datos centralizada. Alojará y ejecutará una aplicación servidora que presentará las páginas web desde las cuales el usuario puede interactuar con el sistema a través de un navegador web, alojará la respectiva lógica del negocio de las funcionalidades y los web services RESTful que permitirán acceder a la lógica del negocio de la captación de pedidos y de clientes desde la

aplicación móvil a través de peticiones y respuestas REST. Las funcionalidades que serán implementadas dentro de este componente permitirán llevar acabo la gestión de los pedidos, el procesamiento y generación de facturas, de guías de remisión y boletas de manera automatizada, la gestión de clientes, la gestión de precios de los productos según clientes y la generación de algunos reportes que permita agilizar y mejorar la toma de decisiones y el servicio que brinda esta empresa.

**Lado del cliente:** Estará formado por la aplicación para móviles. Éste se encarga de realizar peticiones REST al servidor y de procesar respuestas REST del servidor para poder completar la funcionalidad de captación de información de los clientes y de pedidos. Dentro de la lógica de esta aplicación cliente ligero, se implementará las invocaciones a los web services, y el tratamiento de los datos en el formato adecuado para el envío y presentación de los datos sobre su interfaz de usuario que se envíen o se reciban de la aplicación servidora

El protocolo de comunicación entre el cliente y el servidor es el HTTP El formato de los mensajes REST intercambiados será el uso del formato JSON.

En este capítulo se expuso los puntos referentes al aporte teórico de la solución.

En el siguiente capítulo se expondrá lo referente al aporte práctico, donde se abordará el diseño y la implementación de la solución.

## CAPÍTULO V. APOORTE PRÁCTICO

Dentro de este capítulo la primera sección contiene el diseño de la solución planteada, basándonos en la metodología RUP. En el punto 5.1.1 Se presentan los requerimientos funcionales y no funcionales, el catálogo de actores, los casos de uso organizados por paquetes, la especificación de los casos de uso del sistema principales, seguidamente en el punto 5.1.2 se muestra la arquitectura del sistema que viene representada por el Diagrama de Paquetes por capas (Vista lógica), el Diagrama de Componentes (Vista de Implementación), el Diagrama de Secuencia de los principales casos de uso del sistema (Vista de proceso), el Diagrama de Despliegue (Vista de despliegue) a continuación se muestra una ilustración que permite apreciar mejor la arquitectura del sistema en una vista de despliegue y el diagrama de clases del sistema. En el punto 5.1.3 se presenta el diagrama de base de datos. Finalmente, en el paso 5.1.4 se presenta las interfaces gráficas del sistema.

### 5.1 Diseño de la solución

#### 5.1.1 Requerimientos Funcionales y casos de uso del sistema

##### Requerimientos funcionales para el componente web

En la tabla 5.1 se presenta los requerimientos funcionales del módulo Facturación y pedidos.

MÓDULO	FACTURACIÓN Y PEDIDOS
NRO	FUNCIONALIDAD
1	El sistema debe permitir el registro de pedidos en línea en una base de datos centralizada por medio de un navegador web. Para esto es necesario que se seleccione un cliente o se registre en el caso de que no exista y se agregue los productos y las cantidades solicitadas de los productos a sus pedidos.
2	El sistema debe permitir registrar la información de las cantidades que se van a despachar para cada cantidad solicitada de los productos que conforman un determinado pedido.
3	El sistema debe permitir facturar los pedidos que hayan sido registrados con su peso de despacho y realizar los cálculos del importe por cada producto, el cálculo del igv, el subtotal y el total de dinero a pagar.
4	El sistema debe permitir la emisión de boletas para un determinado pedido que haya sido registrado con su peso de despacho, y realizar los cálculos del importe y del total.
5	El sistema debe permitir, la emisión de guías de remisión remitentes para informar el traslado de diversos productos por motivo de venta, para lo cual tomará como entradas información de los clientes y comprobantes brindados por el sistema.
6	El sistema debe permitir a través del componente web la consulta del stock de productos y la información de cada uno de ellos como son su nombre, unidad de medida y las cantidades actualmente disponibles para la venta.

**Tabla 5.1** Requerimientos funcionales del módulo Facturación y pedidos (Fuente: Elaboración propia)

En la tabla 5.2 se muestra los requerimientos funcionales del módulo Gestión.

<b>MÓDULO</b>	<b>Gestión</b>
<b>NRO</b>	<b>FUNCIONALIDAD</b>
1	El sistema debe permitir la agregación de la información de un nuevo producto como lo es su nombre, cantidad actual disponible en el stock, la actualización automática y manual del stock, modificación y la eliminación.
2	El sistema debe permitir crear, eliminar, modificar la información de los clientes.
3	El sistema debe permitir registrar grupos de precios por clientes.
4	El sistema debe permitir agregar, modificar y eliminar la información de los transportistas.
5	El sistema debe permitir a ingresar los datos de la empresa que usará el sistema, además de la tasa de interés del igv con el cual trabajará este sistema.

**Tabla 5.2 Requerimientos funcionales del módulo Gestión (Fuente: Elaboración propia)**

En la tabla 5.3 se muestra los requerimientos funcionales del módulo Reportes:

<b>MÓDULO</b>	<b>Reportes</b>
<b>NRO</b>	<b>FUNCIONALIDAD</b>
1	El sistema debe permitir emitir el reporte que muestre el comportamiento de las ventas de productos en los últimos meses.
2	El sistema debe permitir emitir el reporte de las ventas por cliente en un determinado periodo.

**Tabla 5.3 Requerimientos funcionales del módulo Reportes (Fuente: Elaboración propia)**

En la tabla 5.4 se muestra los requerimientos funcionales del módulo Seguridad:

<b>MÓDULO</b>	<b>SEGURIDAD</b>
<b>NRO</b>	<b>FUNCIONALIDAD</b>
1	El sistema debe permitir la validación de usuario mediante usuario y contraseña al momento de iniciar sesión en el sistema. Además de poder permitirle cerrar sesión.
2	El sistema debe permitir agregar, modificar y eliminar la información de los vendedores que usarán los dispositivos móviles para el sistema, proveídos de un usuario y una contraseña.

**Tabla 5.4 Requerimientos funcionales del módulo Seguridad (Fuente: Elaboración propia)**

### Requerimientos funcionales para el componente móvil

En la tabla 5.5 se detallan los requerimientos funcionales para el componente móvil.

MÓDULO	CLIENTES PEDIDOS Y SEGURIDAD
NRO	FUNCIONALIDAD
1	El sistema debe permitir el registro de pedidos en línea en una base de datos centralizada a través de dispositivos móviles. Para esto es necesario que se seleccione un cliente o se registre en el caso de que no exista y se agregue los productos y las cantidades solicitadas de los productos a sus pedidos.
2	El sistema debe permitir en el registro de los datos de un determinado cliente.
3	El sistema debe permitir la validación de usuario mediante usuario y contraseña al momento de iniciar sesión en el sistema.

**Tabla 5.5 Requerimientos funcionales del módulo Clientes, Pedidos y Seguridad**

(Fuente: Elaboración propia)

### Requerimientos no funcionales del sistema

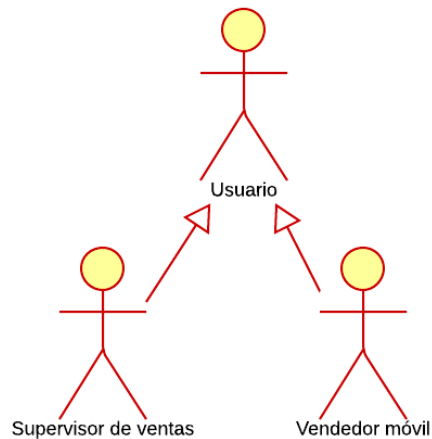
En la tabla 5.6 se muestran los requerimientos no funcionales para el sistema.

REQUERIMIENTOS NO FUNCIONALES	
NRO	REQUERIMIENTO
1	El sistema debe constar de dos componentes: un componente servidor basado en web que permita realizar las funcionalidades de facturación, gestión de pedidos, clientes, productos, así como administración de los usuarios del sistema; y un componente cliente para dispositivos móviles que permita el registro de clientes y de pedidos a través del internet en una base de datos centralizada alojada en el servidor.
2	El componente móvil permitirá el inicio de sesión al sistema una vez verificada la disponibilidad de la base de datos centralizada y de la conexión a internet.
3	El componente web lanzará un aviso que informe que un pedido o un cliente han sido registrados desde el componente móvil, para su atención.
4	El componente móvil debe de poder ser lanzado como una aplicación nativa y no con navegador.
5	El componente móvil debe poder desplegarse en dispositivos móviles con distintos sistemas operativos.

**Tabla 5.6 Requerimientos no funcionales (Fuente: Elaboración propia)**

## Actores

En el gráfico 5.1 se muestran los actores del sistema:



**Figura 5.1 Actores del sistema (Fuente: Elaboración propia)**

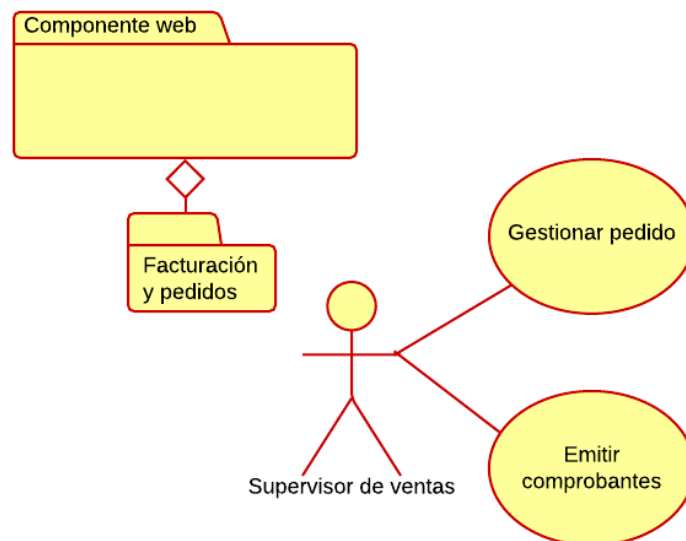
## Casos de uso del sistema por paquetes

### Componente web

A continuación, se muestran los diagramas de caso de uso dispuestos por paquetes.

### Paquete de Facturación y pedidos

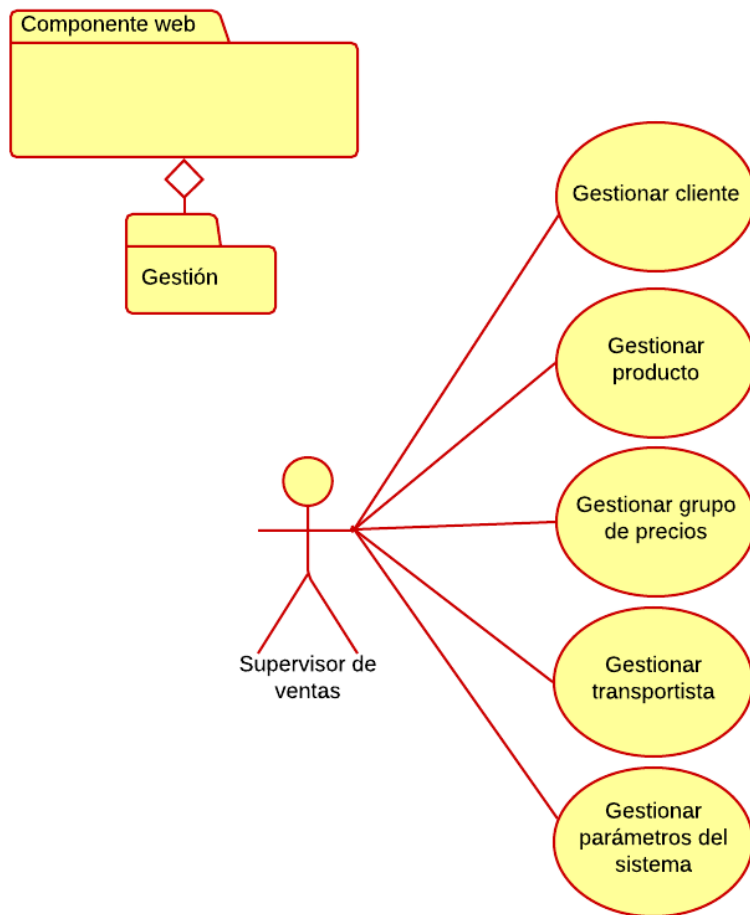
En la figura 5.2 se presentan el diagrama de caso de uso del paquete Facturación y pedidos.



**Figura 5.2 Diagrama de casos de uso del paquete Facturación y pedidos (Fuente: Elaboración Propia)**

### Paquete de Gestión

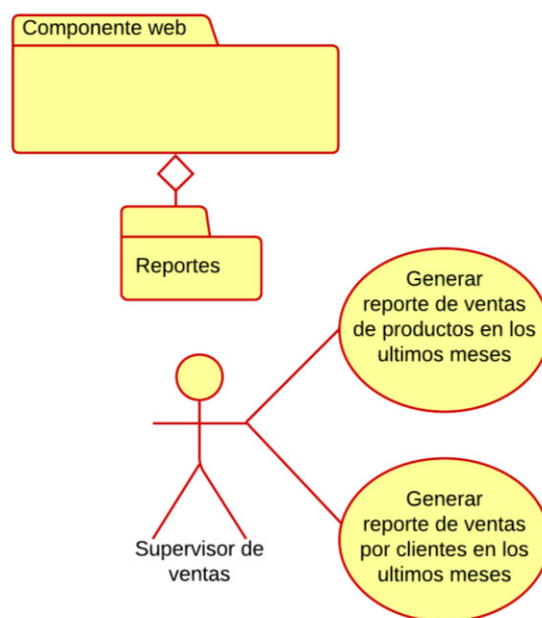
En la figura 5.3 se presentan el diagrama de caso de uso del paquete de Gestión.



**Figura 5.3 Diagrama de casos de uso del paquete Gestión (Fuente: Elaboración Propia)**

### Paquete de Reportes

En la figura 5.4 se presentan el diagrama de caso de uso del paquete de Reportes:

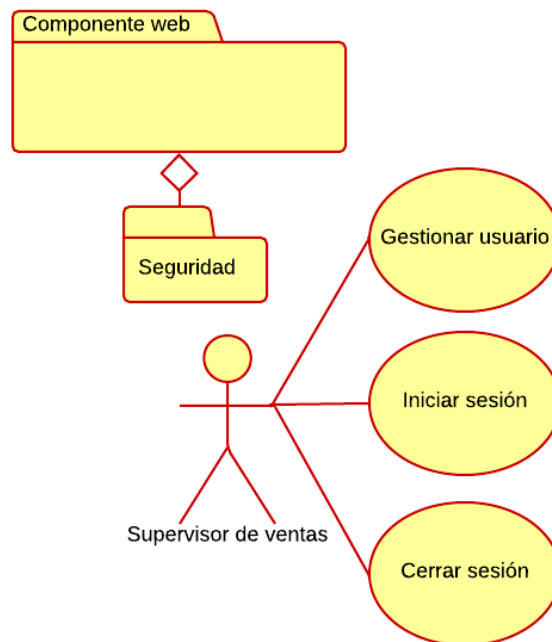


**Figura 5.4 Diagramas de casos de uso del módulo de reportes (Fuente: Elaboración propia)**



### Paquete de Seguridad

En la figura 5.5 se presentan el diagrama de caso de uso del paquete de Seguridad:



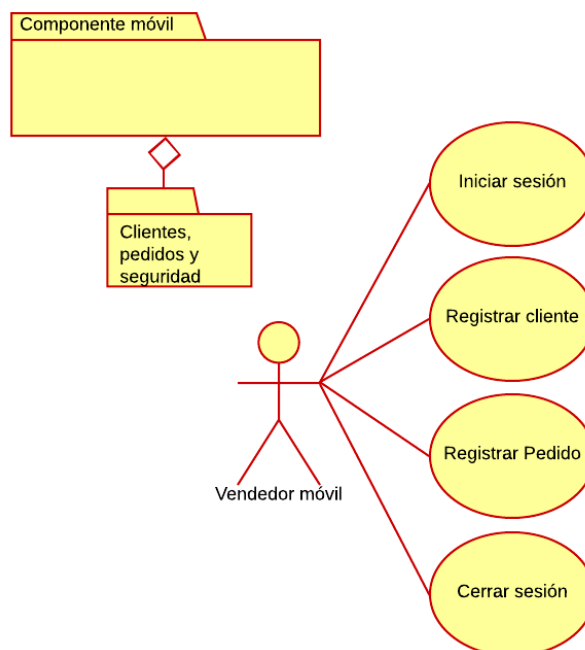
**Figura 5.5 Diagrama de casos de uso del módulo Seguridad (Fuente: Elaboración Propia)**

### Componente móvil

Seguidamente se muestra el diagrama de casos de uso por paquetes para el componente móvil:

### Paquete de clientes pedidos y seguridad

En la figura 5.6 se presentan el diagrama de caso de uso del paquete Clientes, pedidos y seguridad.



**Figura 5.6 Diagrama de casos de uso del módulo Clientes, pedidos y seguridad (Fuente: Elaboración propia)**

## Especificación de los casos de uso principales

En el siguiente apartado se presentan la especificación de los principales casos de uso del sistema:

### Gestionar cliente-Paquete Gestión

En la tabla 5.7 se muestra la especificación del caso de uso gestionar cliente:

<b>ID:</b>	USC01
<b>Caso de Uso:</b>	Gestionar cliente
<b>Actor:</b>	Supervisor de ventas
<b>Descripción:</b>	Permite consultar, registrar, editar y eliminar un determinado cliente.
<b>Precondición:</b>	El Supervisor de ventas debe haber iniciado sesión.
<b>Flujo Principal: Consultar clientes</b>	
<ol style="list-style-type: none"><li>1. El Supervisor de ventas selecciona la opción “Gestionar Cliente”.</li><li>2. El sistema muestra las opciones de “Agregar”, “Editar” y “Eliminar” cliente, así como la lista de clientes disponibles con una opción para “Ver” la información de cada cliente: Tipo de persona; DNI, nombre, apellido paterno, apellido materno (persona natural); RUC, razón social (persona jurídica), distrito, dirección, teléfono, grupo de precios de productos al que pertenece y una opción para “Ver” la información de cada cliente. Provee los filtros: tipo de persona (persona natural y persona jurídica); DNI, nombres, apellido paterno y apellido materno (en el caso de personas naturales); RUC, razón social (en el caso de personas jurídicas); el distrito y grupo de precios de productos al que pertenece el cliente.</li><li>3. El Supervisor de ventas realiza la búsqueda mediante cualquiera de los filtros presentados por el sistema.</li><li>4. El sistema realiza el filtrado de los clientes correspondientes a los filtros aplicados.</li><li>5. El Supervisor de ventas seleccionar el cliente y presiona “Ver”.</li><li>6. El sistema presenta la información completa del cliente.</li></ol>	
<b>Post-condición:</b>	Se consultó los datos del cliente.
<b>Flujo Alternativo: Registrar cliente</b>	
<ol style="list-style-type: none"><li>2.A.1. El Supervisor de ventas selecciona la opción “Agregar”.</li><li>2.A.2. El sistema muestra la interfaz de “registrar cliente”, con las opciones para selección de tipo de cliente, una entrada para el DNI, nombres, apellido paterno, apellido materno, (persona natural); razón social, RUC (persona jurídica); distrito, dirección, teléfono 1, teléfono 2, y un selector para el grupo de precios de productos al cual pertenecerá el cliente.</li><li>2.A.3. El Supervisor de ventas realiza el ingreso de los datos presentados y selecciona “registrar cliente”.</li><li>2.A.4. El sistema presenta un mensaje de operación satisfactoria “Cliente registrado”.</li></ol>	
<b>Post-condición:</b>	Se registró los datos del cliente.
<b>Flujo Alternativo: Editar cliente</b>	

2.B.1. El Supervisor de ventas selecciona el cliente y luego la opción “Editar”.	
2.B.2. El sistema muestra la interfaz de “editar cliente” con una entrada para el DNI, nombres, apellido paterno, apellido materno (persona natural); razón social, RUC (persona jurídica); un selector de distrito, dirección, teléfono 1, teléfono 2, y un selector para el grupo de precios de productos al cual pertenecerá el cliente.	
2.B.3. El Supervisor de ventas realiza el cambio de cualquiera de los datos anteriormente presentados y selecciona “Editar cliente”.	
2.B.4. El sistema presenta un mensaje de operación satisfactoria “Cliente editado”.	
<b>Flujo Alternativo: Eliminar cliente</b>	
2.C.1. El Supervisor de ventas selecciona el cliente y luego “Eliminar”.	
2.C.2. El sistema presenta un mensaje para la confirmación de la eliminación.	
2.C.3. El Supervisor de ventas selecciona aceptar.”	
2.C.4. El sistema realiza la eliminación del cliente y muestra un mensaje de la operación satisfactoria.	
2.C.5. El Supervisor selecciona aceptar.	
<b>Post-condición:</b>	Se eliminó los datos del cliente.

**Tabla 5.7 Especificación del caso de uso Gestionar cliente (Fuente: Elaboración propia)**

## Gestionar pedido-Paquete Facturación y pedidos

En la tabla 5.8 se muestra la especificación del caso de uso Gestionar pedido:

<b>ID:</b>	USC02
<b>Caso de Uso:</b>	Gestionar pedido
<b>Actor:</b>	Supervisor de ventas
<b>Descripción:</b>	Permite gestionar los pedidos de los clientes.
<b>Precondición:</b>	La aplicación debe haber sido iniciada correctamente.
<b>Flujo Principal: Consultar pedido</b>	
<ol style="list-style-type: none"><li>1. El Supervisor de ventas selecciona la opción “Gestionar Pedidos”.</li><li>2. El sistema muestra las opciones de “Agregar”, “Editar” y “Eliminar” pedido, así como la lista de pedidos disponibles con una opción para “Ver” la información de cada pedido, una opción para ver el “Comprobante” (factura o boleta) y otra opción para ver la “Guía de remisión” si es que éstos han sido emitidos para el pedido. Provee los filtros: tipo de persona (persona natural y persona jurídica), filtro de comprobante (no facturado y facturado), opciones de búsqueda por fecha (fecha de entrega, fecha de registro y fecha de facturación), dos entradas para la introducción del rango de fechas para este tipo de búsqueda y el tipo de entregas del pedido (si es delivery o no). Adicionalmente el sistema presenta filtros para la búsqueda de los pedidos mediante el nombre del cliente, apellido paterno y materno (en caso de personas naturales) o la razón social (en caso de personas jurídicas). Así como opciones de ordenamiento de pedidos dentro de la lista para el Id del pedido, fecha de realización y la fecha de entrega. También presenta las opciones para aplicar los filtros en la búsqueda y otra opción para limpiar los filtros que se hayan utilizado en la búsqueda.</li><li>3. El Supervisor de ventas realiza la búsqueda mediante cualquiera de los filtros presentados por el sistema.</li><li>4. El sistema realiza el filtrado de los pedidos correspondientes a los filtros aplicados.</li><li>5. El Supervisor de ventas seleccionar el pedido y presiona “Ver”.</li><li>6. El sistema presenta la información: la fecha de realización, la fecha de entrega, la información del tipo de entrega, los detalles (nombres de productos con la cantidad solicitada), los subtotales para cada detalle, el precio total y el comentario del pedido.</li></ol>	
<b>Post-condición:</b>	Se consultó la información del pedido.
<b>Flujo Alternativo: Registrar pedido</b>	

2.A.1. El Supervisor de ventas selecciona la opción “Agregar”.	
2.A.2. El sistema muestra la interfaz de “registrar pedido”	
2.A.3. El Supervisor de ventas selecciona “ingresar cliente”.	
2.A.4. El sistema muestra la lista de los clientes con los filtros: tipo de persona, DNI, Nombre y Apellidos para realizar la búsqueda del cliente registrado. Si es un nuevo cliente se realizará el registro del cliente a través del flujo alternativo “Registrar cliente” del caso de uso “Gestionar cliente”.	
2.A.5. El Supervisor de ventas realiza el filtrado de los clientes por cualquiera de los filtros presentados por el sistema. Una vez localizado el cliente, el Supervisor de ventas selecciona el cliente y presiona “Seleccionar cliente”.	
2.A.6. El sistema oculta la lista de los clientes y carga el nombre del cliente y los productos con los precios correspondientes a dicho cliente y la unidad de medida del producto.	
2.A.7. El Supervisor de ventas selecciona la fecha de registro, la fecha de entrega del pedido y una opción para marcar el tipo de entrega del pedido (delivery o no).	
2.A.7.1. Realiza la búsqueda del producto en la lista de productos disponibles, selecciona el producto y a continuación selecciona añadir.	
2.A.8. El sistema muestra una caja de diálogo con el nombre del producto, la unidad de medida de éste y un campo de texto para el ingreso de la cantidad solicitada por el cliente.	
2.A.9. El Supervisor de ventas ingresa la cantidad del producto y selecciona aceptar.	
2.A.10. El sistema agrega el producto seleccionado a la lista de productos del pedido y calcula el subtotal y el precio total del pedido de acuerdo a la cantidad del producto y el precio unitario del producto para el cliente.	
2.A.11. Se repite la secuencia de pasos desde el paso 2.A.7.1 hasta el paso 2.A.10, tantas veces como productos del pedido se requieran.	
2.A.12. El Supervisor de ventas ingresa un comentario para el pedido si es que existe la necesidad algo adicional.	
2.A.13. El Supervisor de ventas selecciona registrar pedido.	
2.A.14. El sistema muestra el mensaje de “pedido registrado”.	
<b>Post-condición:</b>	Se modificó los datos del pedido.
<b>Flujo Alternativo: Editar pedido</b>	

2.B.1. El supervisor selecciona la opción “Editar”.	
2.B.2. El sistema presenta los datos del pedido: la fecha de realización, la fecha de entrega, la información del tipo de entrega, los detalles (nombres del producto con la cantidad solicitada) y el comentario del pedido.	
2.B.3. El supervisor de ventas realiza la modificación de cualquiera de estos datos. Para realizar la modificación de los detalles de los pedidos, el supervisor selecciona los botones de “editar” o “eliminar” para cada detalle según lo requerido. Finalmente selecciona modificar.	
2.B.4. El sistema muestra el mensaje de modificación satisfactoria.	
<b>Post-condición:</b>	Se modificó el pedido.
<b>Flujo Alternativo: Eliminar pedido</b>	
2.C.1. El Supervisor de ventas selecciona el pedido y luego Eliminar.	
2.C.2. El sistema presenta un mensaje para la confirmación de la eliminación.	
2.C.3. El Supervisor de ventas selecciona aceptar.	
2.C.4. El sistema realiza la eliminación del pedido y muestra un mensaje de la operación satisfactoria.	
2.C.5. El Supervisor selecciona aceptar.	
<b>Post-condición:</b>	Se eliminó el pedido

**Tabla 5.8 Especificación del caso de uso Gestionar pedido (Fuente: Elaboración propia)**

## Emitir comprobantes-Paquete Facturación y pedidos

En la tabla 5.9 se muestra la especificación del caso de uso Emitir comprobantes:

<b>ID:</b>	USC03
<b>Caso de Uso:</b>	Emitir comprobantes
<b>Actor:</b>	Supervisor de ventas
<b>Descripción:</b>	Permite emitir las facturas, boletas o guías de remisión remitente para los pedidos.
<b>Precondición:</b>	La aplicación debe haber sido iniciada correctamente.
<b>Flujo Principal: Emitir Comprobantes</b>	
<ol style="list-style-type: none"> <li>1. El Supervisor de ventas selecciona la opción “Emitir comprobante”.</li> <li>2. El sistema muestra la interfaz de “emitir comprobante”.</li> <li>3. El Supervisor de ventas selecciona la fecha de entrega y luego selecciona “cargar pedidos pendientes”.</li> <li>4. El sistema presenta la lista de los pedidos pendientes de facturación (falta de boleta, factura o guía de remisión remitente ya sea en el caso de que alguno de los comprobantes respectivos no hayan sido emitidos o anulados) con los datos: código del pedido, nombre del cliente, tipo de cliente, fecha de registro, fecha de entrega, tipo de entrega, nombre del vendedor, canal (registro a través del móvil o de la web del sistema), un campo para informar si se han registrado las cantidades despachadas para los pedidos, así como las opciones para ver el comprobante de pago o guía de remisión remitente emitidos. También presenta los filtros para: el nombre del cliente, el tipo de cliente (persona natural y persona jurídica), tipo de entrega y para cantidades de despacho registradas. Además, presenta opciones de ordenamiento para la fecha de registro y para la fecha de entrega.</li> <li>5. El Supervisor de ventas busca el pedido a través de los filtros disponibles; a continuación, selecciona el pedido para la asignación de las cantidades de despacho. Y selecciona “registrar cantidades de despacho”.</li> <li>6. El sistema carga los detalles de pedido (nombre de producto, cantidad original y cantidad de despacho).</li> <li>7. El Supervisor de ventas ingresa la cantidad de despacho para cada detalle de pedido y finalmente selecciona “registrar”, hasta terminar el llenado de las cantidades de todos los detalles del pedido.</li> <li>8. El sistema indica el registro satisfactorio de las cantidades de despacho y marca al pedido con el campo de cantidades de despacho asignadas.</li> <li>9.1 El Supervisor de ventas selecciona el pedido.</li> <li>9.2 El supervisor presiona “Emitir comprobantes”</li> <li>10. El sistema verifica si algún pedido es de delivery entonces muestra un formulario con los datos necesarios para la guía de remisión: punto de partida, razón de la empresa de transporte, ruc de la empresa de transporte, vehículo de transporte, constancia de inscripción y el motivo de transporte.</li> <li>11. El supervisor de ventas selecciona o ingresa los datos correspondientes, y luego presiona “emitir comprobantes”.</li> <li>12. El sistema verifica el tipo de cliente para cada pedido, seguidamente genera la factura o boleta en caso de tener o no RUC el cliente respectivamente. Además, se genera la guía de remisión en el caso de que algún pedido sea con entrega a domicilio. A continuación, el sistema presenta un mensaje de emisión satisfactoria de comprobantes.</li> <li>13. Luego de haber registrado las cantidades a despachar para “n” pedidos (“n” veces desde el punto 5 hasta el punto 9.1). El supervisor de ventas selecciona los “n” pedidos y presiona “Emitir comprobantes”. Seguidamente se lleva a cabo desde el punto 10 hasta el punto 12.</li> </ol>	
<b>Post-condición:</b>	Se emitió los comprobantes para los pedidos
<b>Flujo Alternativo: Cambiar estado del comprobante</b>	

2.A.1. El Supervisor de ventas selecciona la opción “Cambiar estado del comprobante”.	
2.A.2. El sistema muestra la información de los comprobantes asignados tales como tipo de comprobante (boleta, factura y guía de remisión), número de comprobante, estado (aprobado y anulado), número de fecha de emisión, opciones para aprobar y anular comprobante y los filtros por el tipo de comprobante, estado del comprobante y opciones para ordenamiento por número de comprobante.	
2.A.3. El Supervisor de ventas selecciona la opción anular o aprobar de acuerdo a la disponibilidad del formato del comprobante preimpreso para la realización de impresión de los datos del pedido en formato..	
<b>Post-condición:</b>	Se emitió la boleta o factura o guía de remisión para los pedidos.

**Tabla 5.9 Especificación del caso de uso Emitir comprobante (Fuente: Elaboración propia)**

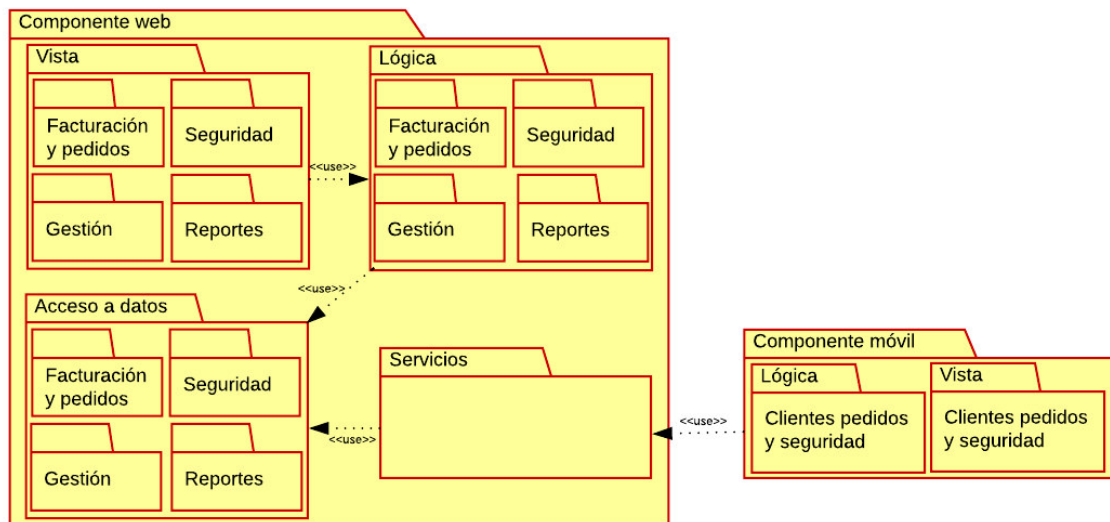


### 5.1.2 Arquitectura del sistema y diagrama de clases

En el siguiente apartado se presentan la arquitectura del sistema, que viene descrita por la vista lógica (diagrama de paquetes), la vista de implementación (diagrama de componentes), la vista de proceso (diagramas de secuencias), y la vista de despliegue (diagrama de despliegue).

#### *Diagrama de Paquetes por capas (Vista lógica)*

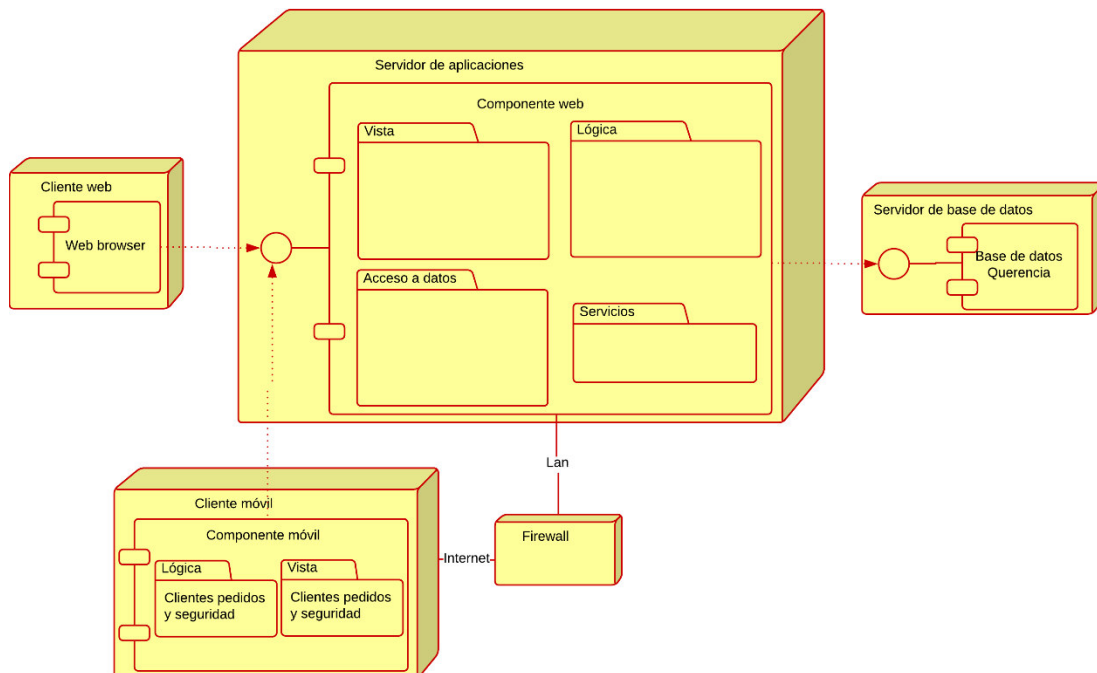
En la figura a continuación se muestra la vista lógica del sistema (diagrama de paquetes).



**Figura 5.7 Diagrama de paquetes del sistema (Fuente: Elaboración propia)**

#### *Diagrama de Componentes (Vista de Implementación)*

En la figura siguiente se muestra la vista de implementación (diagrama de componentes):



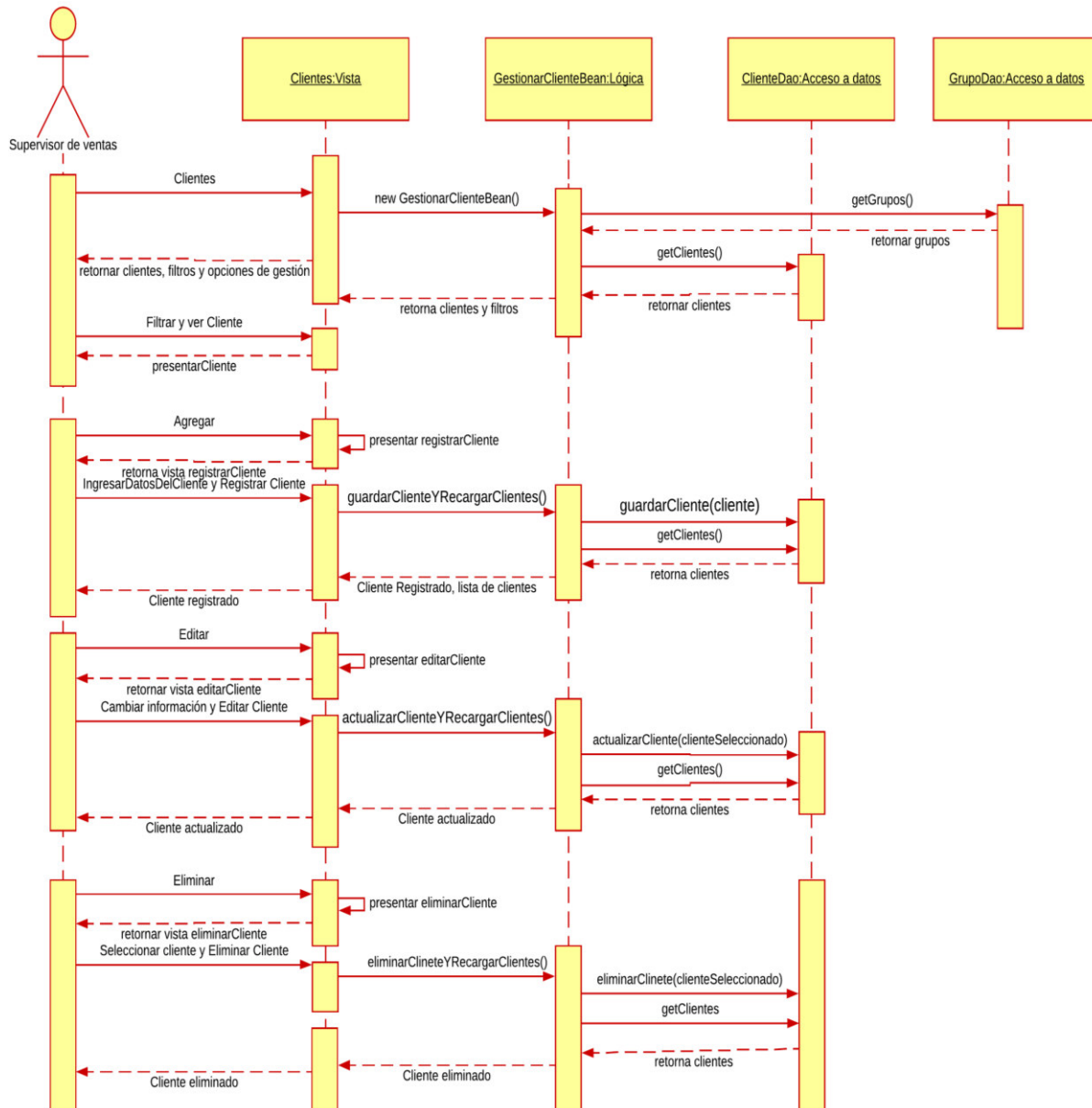
**Figura 5.8 Diagrama de componentes del sistema (Fuente: Elaboración propia)**

## Diagramas de Secuencia (Vista de proceso)

Seguidamente se presenta la vista de proceso del sistema (diagramas de secuencias):

### Diagrama de secuencia del caso de uso de uso Gestionar cliente

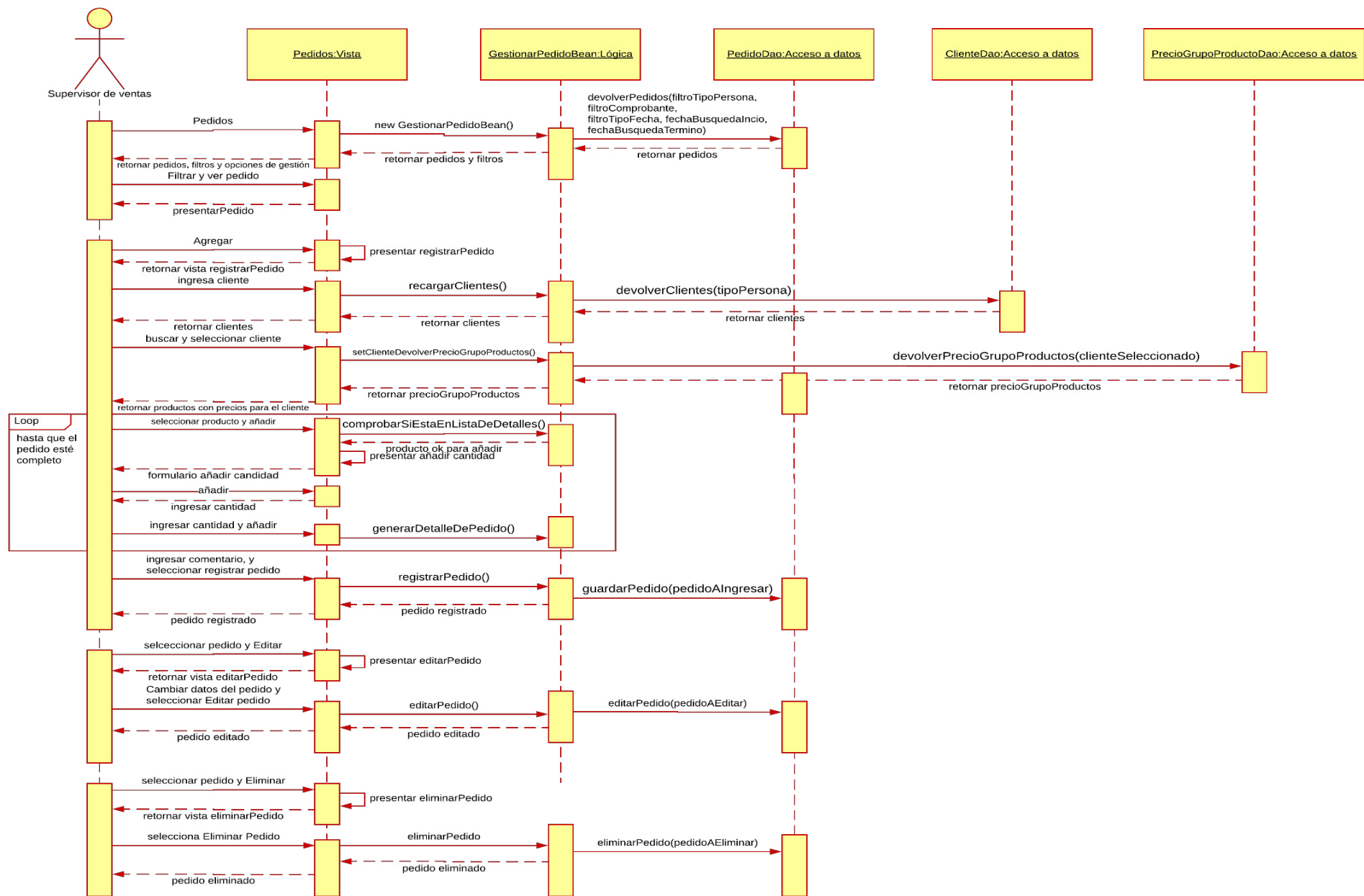
En la figura que sigue se muestra el diagrama de secuencia del caso de uso gestionar cliente:



**Figura 5.9 Diagrama de secuencia Caso de uso Gestionar cliente (Fuente: Elaboración propia)**

### Diagrama de secuencia del caso de uso Gestionar pedido

En la figura a continuación se muestra el diagrama de secuencia del caso de uso Gestionar pedido:



**Figura 5.10 Diagrama de secuencia Caso de uso Gestionar pedidos (Fuente: Elaboración propia)**

Diagrama de secuencia del caso de uso de uso Emitir comprobante

En la figura a continuación se muestra el diagrama de secuencia del caso de uso Emitir comprobante:

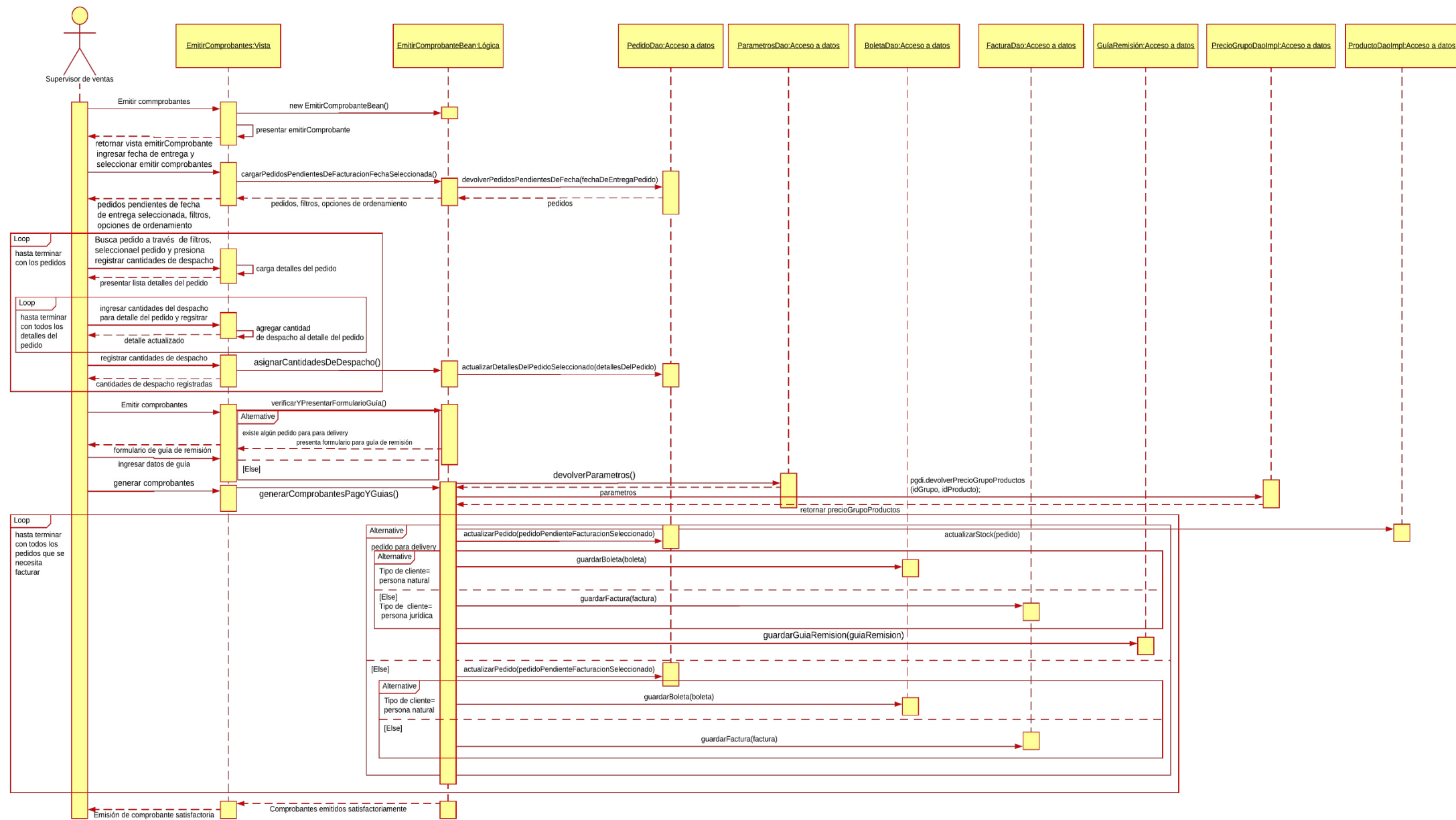
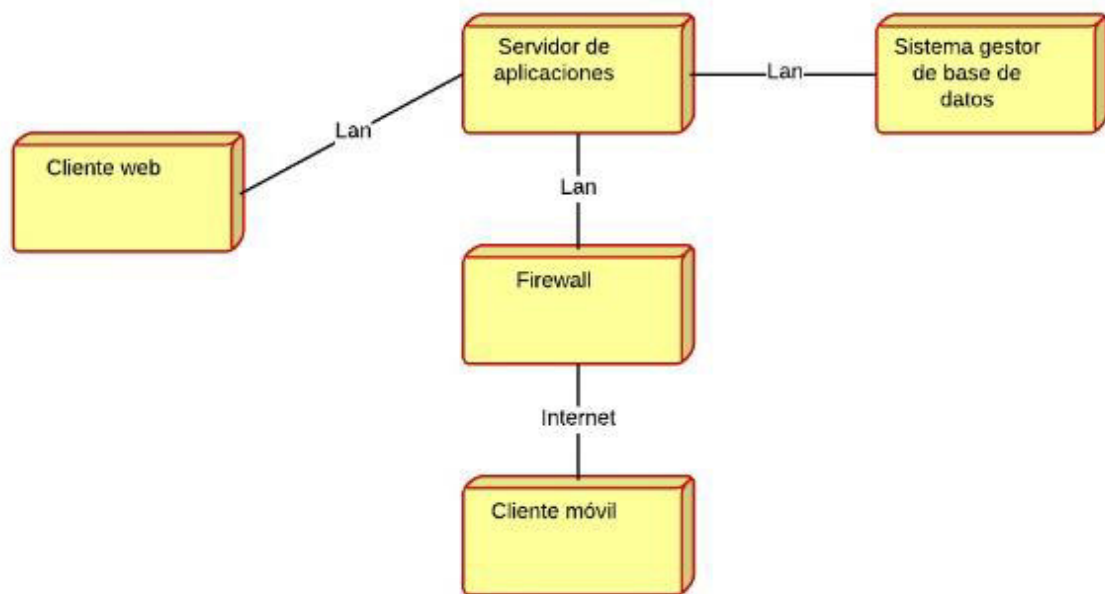


Figura 5.11 Diagrama de secuencia Caso de uso Emitir comprobante (Fuente: Elaboración propia)

**Diagrama de Despliegue (Vista de despliegue)**

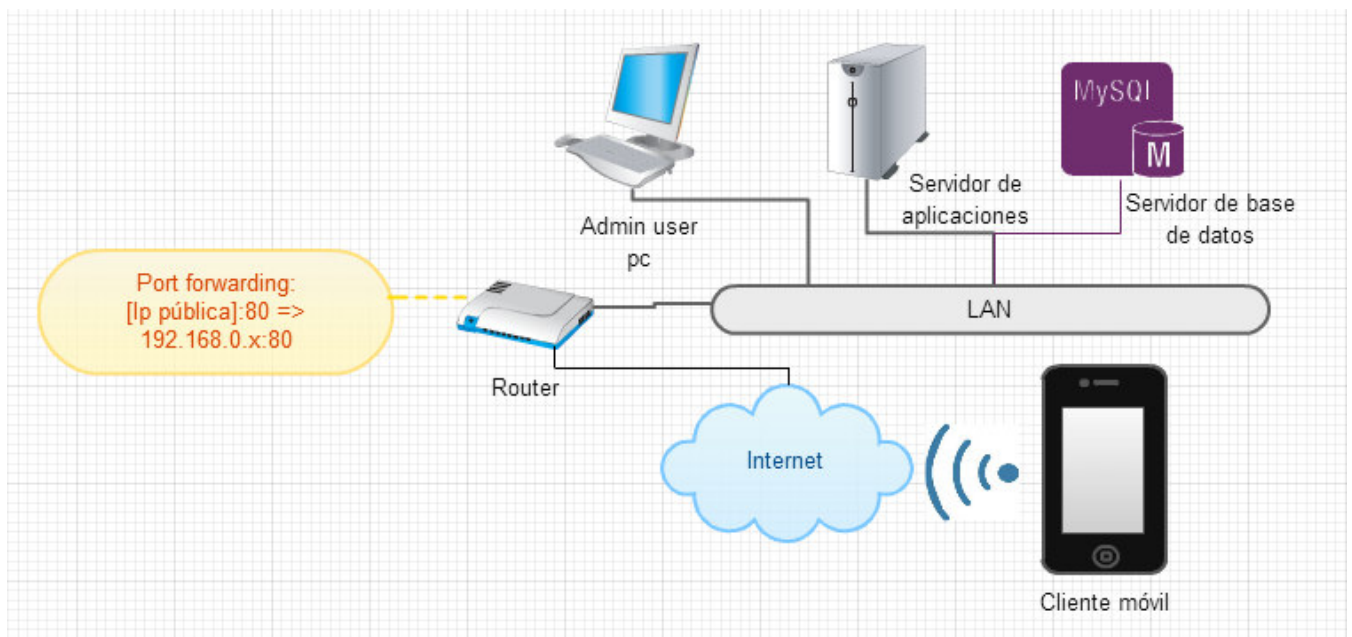
En la figura siguiente se presenta el diagrama de despliegue del sistema.



**Figura 5.12 Diagrama de despliegue del sistema (Fuente: Elaboración propia)**

### Arquitectura del sistema (Vista de despliegue)

En la figura a continuación se puede apreciar otra vista de despliegue del sistema para poder identificar mejor los nodos necesarios y aspectos de la configuración de algunos de ellos para poder establecer la comunicación entre los componentes del sistema:



**Figura 5.13 Diagrama de despliegue ilustrativo del sistema (Fuente elaboración propia)**

La configuración del router (o del firewall que éste tenga) que permitirá la comunicación entre los dispositivos móviles que realizan sus peticiones desde la Internet hacia el servidor de aplicaciones, que se encuentra dentro de la lan debe de realizarse a través de la redirección de puertos (a veces llamada tunelado), que es la acción de redirigir un puerto de red de un nodo de red a otro de un router con NAT activado. En la figura anterior se aprecia que esta configuración es implementada a través del puerto 80 y que redirección a la ip privada que se encuentra en la lan y que corresponde al servidor de aplicaciones y que también tiene el puerto 80 que es el utilizado para las peticiones http.

Además, el router debe de estar habilitado con una ip pública fija (que no cambie conforme pase el tiempo) para poder realizar las peticiones desde el componente móvil, utilizando la ip pública del router y a través del port forwarding mencionado permitiéndole así llegar hasta el servidor de aplicaciones.



## Diagrama de clases

En la figura que sigue se presenta el diagrama de clases del sistema:

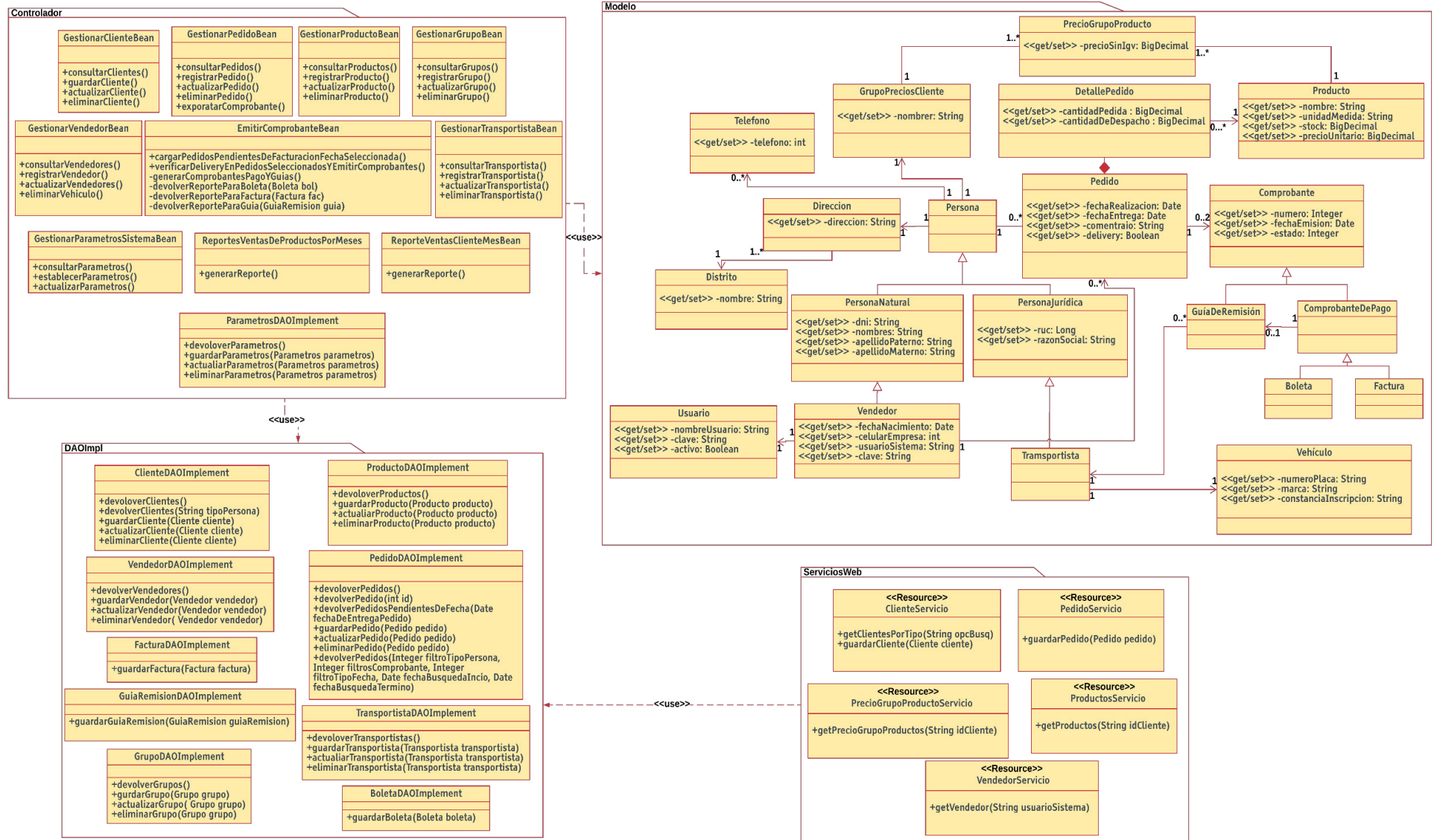


Figura 5.14 Diagramas de clases del sistema (Fuente: Elaboración propia)

### 5.1.3 Diseño de base de datos

En la figura que sigue se presenta el esquema de base de datos del sistema:

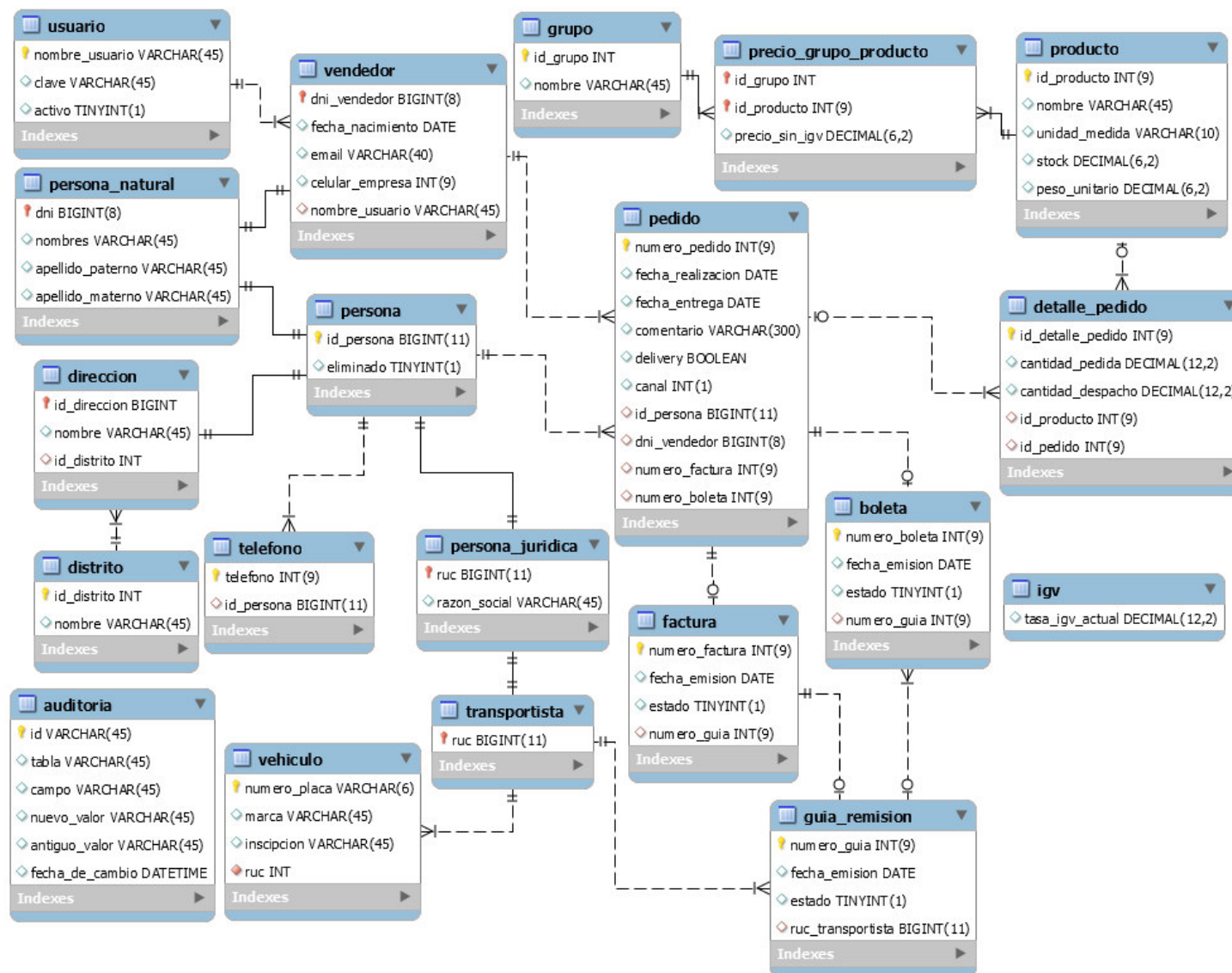


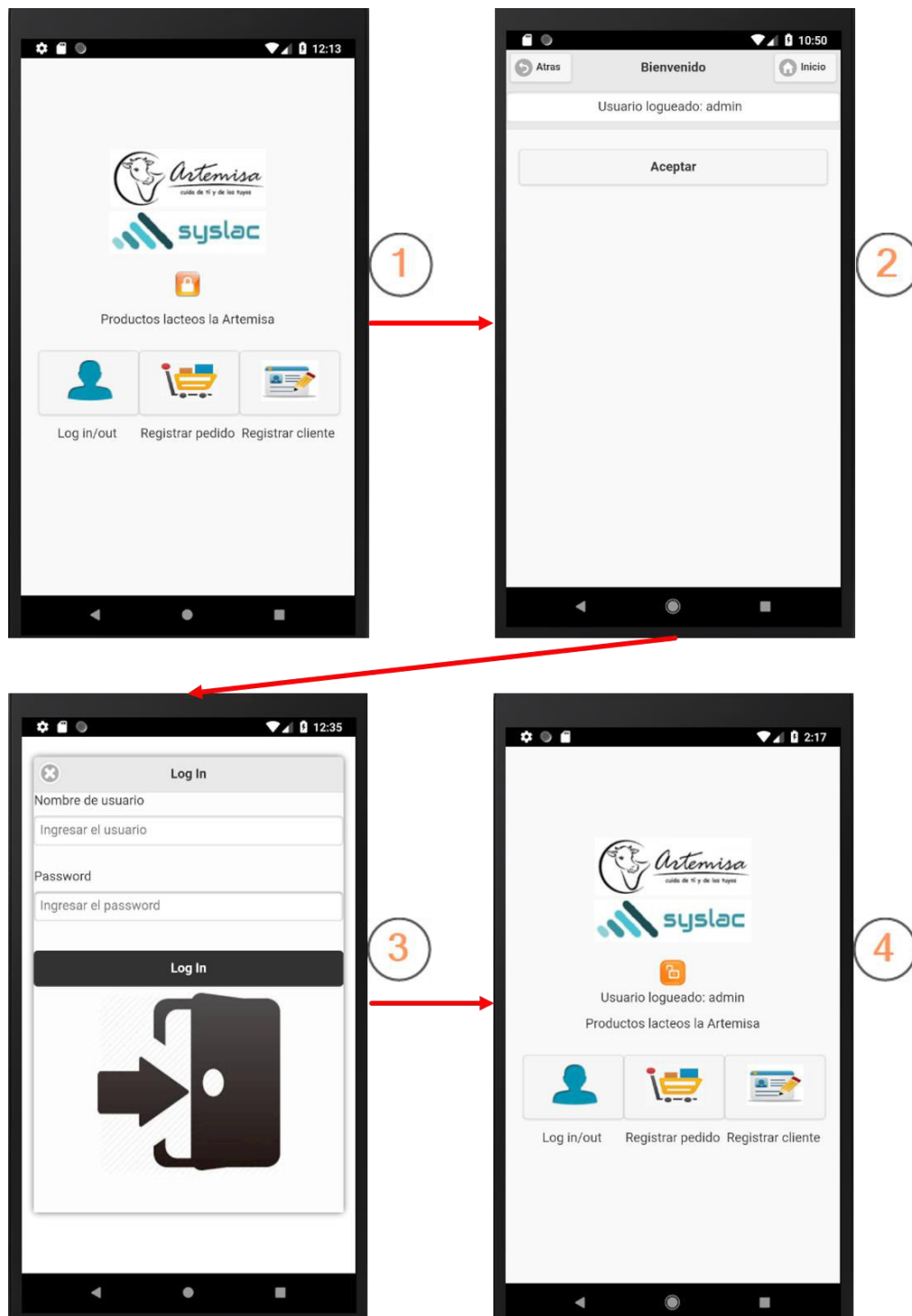
Figura 5.15 Esquema de base de datos del sistema (Fuente: Elaboración propia)



## Prototipado del componente móvil

### 5.1.4 Diseño de interfaces gráficas

En la figura siguiente se muestra el prototipado de las pantallas de bienvenida, y de inicio de sesión:



**Figura 5.16** Prototipo de interfaz de usuario para el caso de uso de iniciar sesión para el dispositivo móvil (Fuente: Elaboración Propia)

La siguiente figura muestra el prototipo de las pantallas del caso de uso registra cliente para el dispositivo móvil:

La siguiente figura muestra el prototipo de las pantallas del caso de uso registra cliente para el dispositivo móvil:

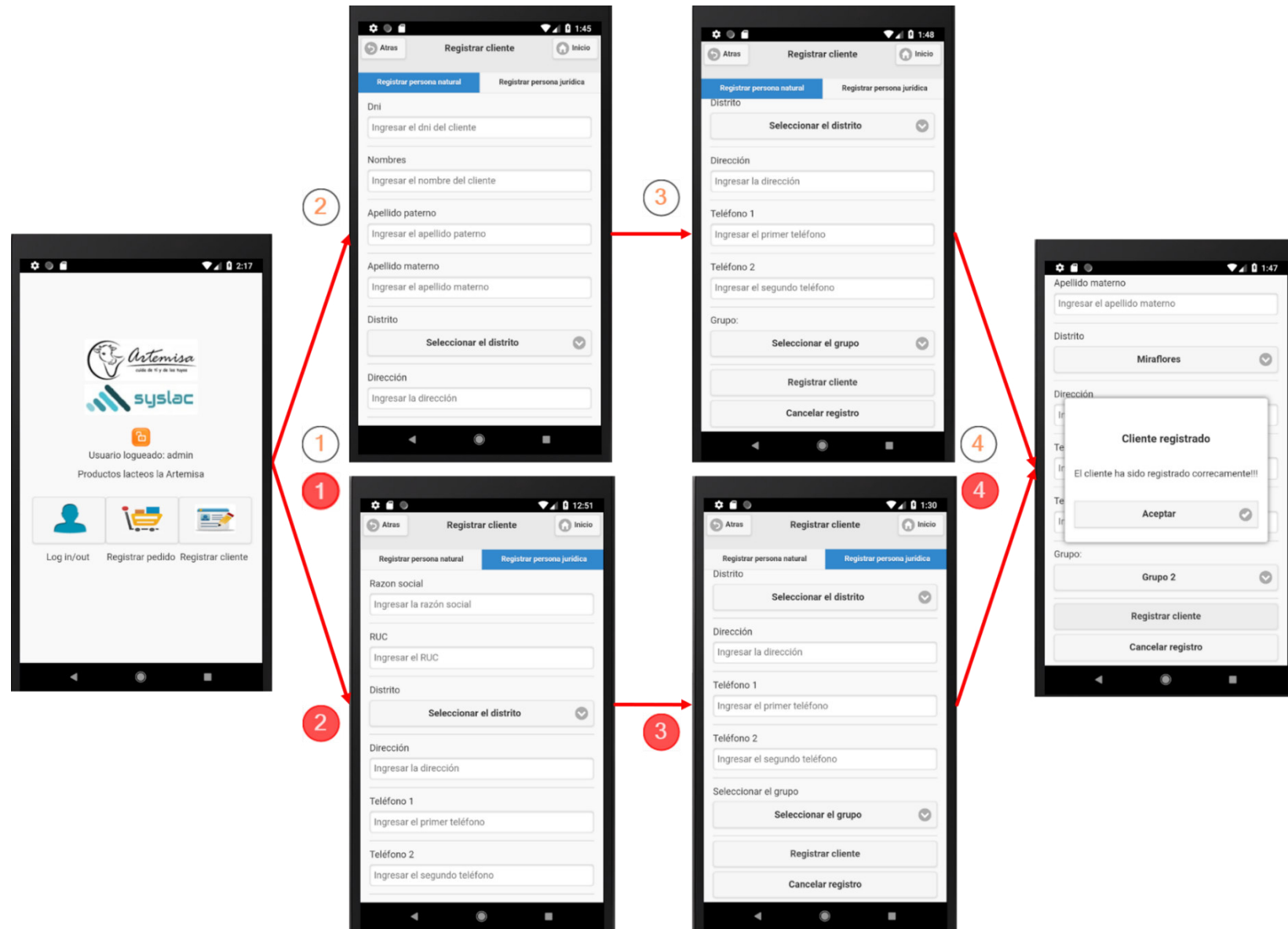


Figura 5.17 Prototipo de interfaz de usuario del caso de uso registra cliente para el dispositivo móvil (Fuente: Elaboración propia)

En la siguiente figura se muestra el prototipo de las pantallas para el caso de uso registrar pedido en el dispositivo móvil:

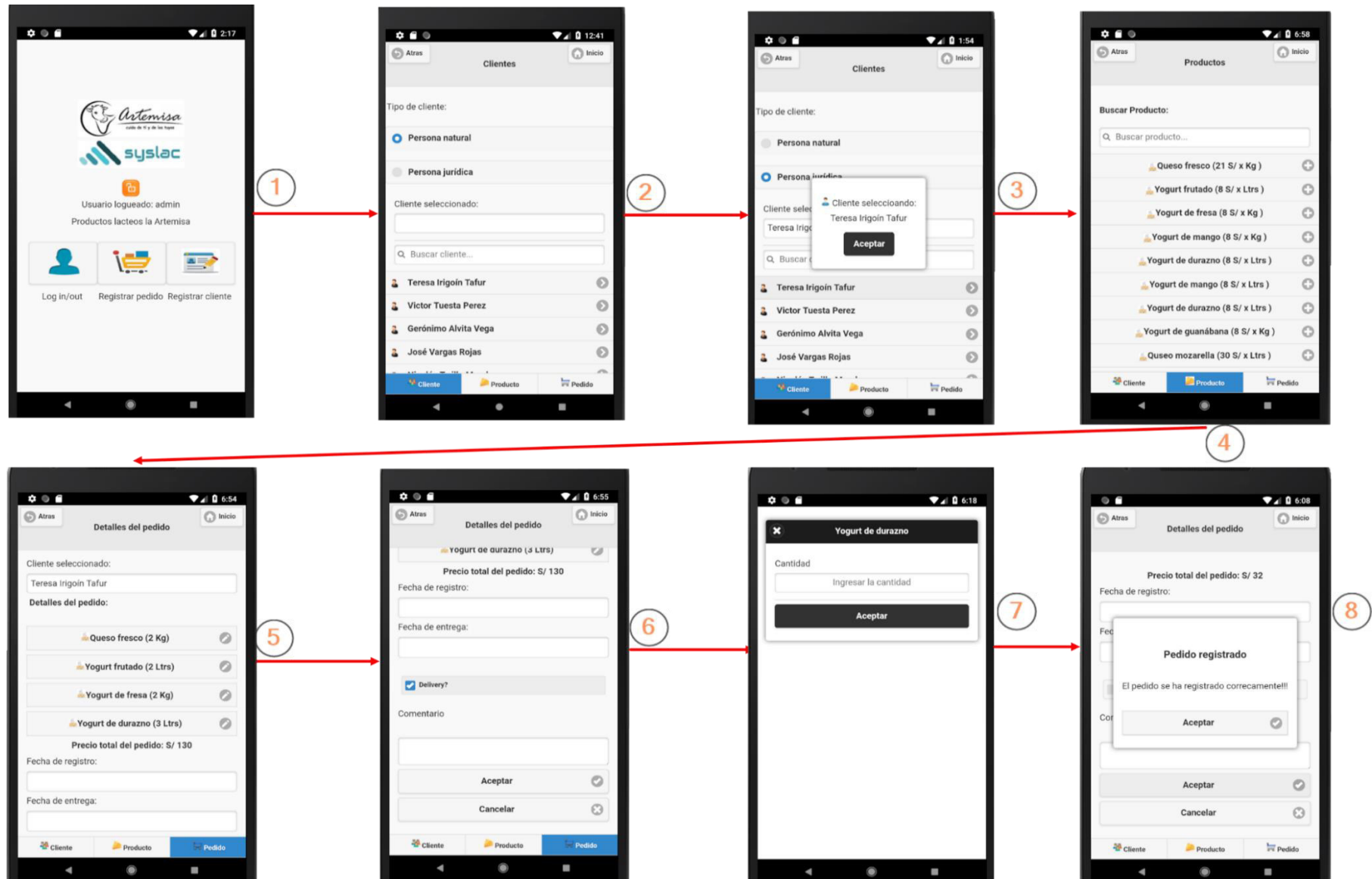


Figura 5.18 Prototipito de interfaz de usuario del caso de uso registra pedido para el dispositivo móvil (Fuente: Elaboración propia)

## Prototipo para componente web

En la siguiente figura se muestra la pantalla de inicio del componente web.



**Figura 5.19 Pantalla de inicio del componente web (Fuente: Elaboración propia)**

En la figura siguiente se muestra la interfaz principal del caso de uso gestionar cliente:



**Figura 5.20 Pantalla del caso de uso de gestión de clientes del componente web (Fuente: Elaboración propia)**

En la siguiente figura se muestra la interfaz principal del caso de uso gestionar pedido.

The main interface for managing orders in the Artemisa web component. It features a header with the Syslac and Artemisa logos, a navigation menu, and a main content area with filters and a table of orders.

**Filters:**

- Filtro por tipo de persona:**
  - Buscar por persona natural
  - Buscar por persona jurídica
- Filtro de comprobante:**
  - No facturado
  - Facturado
- Filtro por fecha:**
  - Búsqueda por fecha:
  - Fecha de entrega
  - Fecha de registro
  - Fecha de facturación
  - Fecha de inicio
  - Fecha de termino
  - Filtrar
  - Limpiar

**Table of Orders:**

Id. del pedido	Nombre	Apellido paterno	Apellido materno	Fecha de realización	Fecha de entrega	Delivery	Vendedor	Canal	Comprobante	Guía de remisión	Ver
628	Asuncion	Gonzales	Gonzales	04/04/2018	05/04/2018		Sheyla Rojas Gonzáles				
705	Sebastian	Vela	Cubas	04/04/2018	07/04/2018		Sheyla Rojas Gonzáles				
753	Asuncion	Gonzales	Gonzales	18/04/2018	19/04/2018		Eliana Guevara Gálvez				
634	Sebastian	Vela	Cubas	04/04/2018	05/04/2018		Sheyla Rojas Gonzáles				
704	Moises	Valera	Cubas	04/04/2018	07/04/2018		Eliana Guevara Gálvez				

Figura 5.21 Interfaz del caso de uso gestionar pedido del componente web (Fuente: Elaboración propia)

En la figura siguiente se muestra la interfaz del flujo registrar pedido del caso de uso Gestionar pedido:

The 'Registrar Pedido' (Register Order) interface in the Artemisa web component. It includes a header, a form for customer and delivery details, a product selection table, and a summary table.

**Form Fields:**

- Ingresar cliente: Miguel Asturias Fuentes
- Fecha del registro de pedido: 25/06/2018
- Fecha de entrega: 26/06/2018
- Entrega por delivery: ☒

**Product Selection Table:**

Producto	Precio unitario	Unidad de medida
Yogurt frutado	S/. 8.00	Ltrs
Queso Mozzarella	S/. 30.00	Kg
Yogurt de fresa	S/. 8.00	Kg
Queso fresco	S/. 21.00	Kg

**Summary Table:**

Producto	Cantidad pedida	Subtotal aproximado
Yogurt frutado	4.000 Ltrs	S/. 32.00
Queso fresco	1.000 Kg	S/. 21.00
Yogurt de fresa	3.000 Ltrs	S/. 24.00
<b>Precio total aproximado:</b>		<b>S/. 77.00</b>

**Comentario:**

Registrar pedido

Figura 5.22 Interfaz del caso de uso registrar pedido del componente web (Fuente: Elaboración propia)

En la siguiente figura se muestra la interfaz del caso de uso emitir componente.





**Artemisa**  
 cuida de tí y de los tuyos



[Inicio](#)
[Gestión](#)
[Facturación y pedidos](#)
[Reportes](#)
[Seguridad](#)
[Ayuda](#)

Logout

Fecha de Entrega:



Pedidos pendientes de facturación												
(1 of 2) [1] [2] [5]												
Emitir comprobantes	Código de pedido	Cliente	Tipo de cliente	Fecha de registro	Fecha de entrega	Entrega delivery?	Vendedor	Canal	Cantidades de despacho ingresadas?	Detalles del pedido	Comprobante	Guía
<input type="checkbox"/>	803	Moisés Valera Cubas	Persona natural	02/04/2018	05/04/2018		Eliana Guevara Gálvez			✓ Asignar cantidades de despacho	<a href="#">Ver comprobante</a>	<a href="#">Ver guía</a>
<input type="checkbox"/>	629	Jhoanna Estela Rojas	Persona natural	04/04/2018	05/04/2018		Sheyla Rojas Gonzáles			✓ Asignar cantidades de despacho	<a href="#">Ver comprobante</a>	<a href="#">Ver guía</a>
<input type="checkbox"/>	628	Alexander Dávila Vásquez	Persona natural	04/04/2018	05/04/2018		Eliana Guevara Vásquez			✓ Asignar cantidades de despacho	<a href="#">Ver comprobante</a>	<a href="#">Ver guía</a>
<input type="checkbox"/>	626	Teresa Irigoin Tafur	Persona natural	04/04/2018	05/04/2018		Sheyla Rojas Gonzáles			✓ Asignar cantidades de despacho	<a href="#">Ver comprobante</a>	<a href="#">Ver guía</a>
<input type="checkbox"/>	624	Jorge Tapia Fuentes	Persona natural	05/04/2018	05/04/2018		Eliana Guevara Gálvez			✓ Asignar cantidades de despacho	<a href="#">Ver comprobante</a>	<a href="#">Ver guía</a>

(1 of 2) [1] [2] [5]

[Emitir comprobantes](#)

**Figura 5.23 Interfaz del caso de uso emitir comprobante del componente web (Fuente: Elaboración propia)**



## 5.2 Implementación

### 5.2.1 Tecnología utilizada para la implementación (Hardware y software)

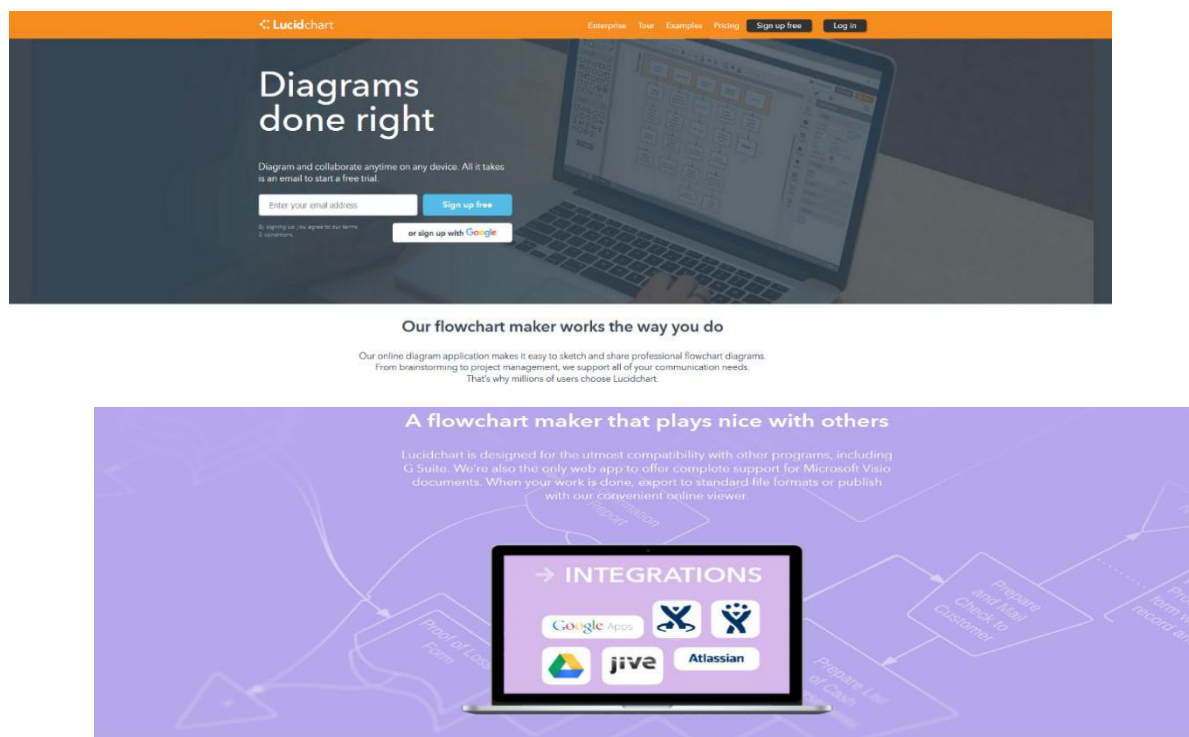
A continuación, se detalla las diferentes herramientas de software utilizadas para el desarrollo de este trabajo de investigación.

#### Software:

##### ❖ LucidChart

De acuerdo a (Lucidchart, 2008), ésta es una aplicación web colaborativa que sirve para realizar diversos tipos de diagramas, entre ellos: diagramas de flujo, mapas mentales, tormenta de ideas, uml, etc. La forma de diagramar es a través de arrastrar y soltar. Esta herramienta es compatible con varios programas incluidos Microsoft Visio, G Suite.

En la figura siguiente se muestra la página web de Lucidchart, como se puede apreciar es un software de paga, pero también puede ser utilizado de manera gratuita por 14 días y nos permitiría almacenar y modificar nuestro diagrama cuando lo queramos, para esto se debe registrar con una cuenta de Gmail para poder gestionar nuestros diagramas.



**Figura 5.24 Ejemplo de diagrama y página web de Lucidchart(Fuente: (Lucidchart, 2008))**

##### ❖ Creately

Creately es un software de diagramación y software de diseño. Es una herramienta basada en la nube. Ofrece funcionalidades que permite diagramar fácil con figuras inteligentes y

conectores inteligentes, permite realizar el trabajo colaborativo y tiene miles de plantillas. Ofrece sus productos de forma online a través de navegadores web, de escritorio y móvil. Además, ofrece plugins para el trabajo conjunto con Confluence, Jira y la G suite (Creately, 2008).

En la siguiente figura se muestra el sitio web del software basado en la web Creately:



**Figura 5.25 Sitio web de Creately (Fuente: (Creately, 2008))**

### ❖ MySQL 5.6 y Mysql Workbench 6.2

MySQL es uno de los sistemas de gestión de bases de datos relacionales más populares (RDBMS). Es de código abierto y está disponible bajo los términos de la Licencia Pública General de GNU, que significa que puede descargar, ejecutar, compartir e incluso modificar MySQL de forma gratuita. MySQL es una elección popular de base de datos para aplicaciones web, y es un componente de los ampliamente utilizados LAMP (Linux, Apache, Mysql/MaríaDB, Perl/PHP/Phyton) (Peicevic, 2016).

### ❖ Netbeans 8.0.1 con soporte para Apache Cordova y para Apache Tomcat 8.0.1



Netbeans es un entorno de desarrollo 100% libre y de código abierto que ayuda a crear diferentes tipos de aplicaciones de software. El IDE brinda soporte para todo tipo de aplicaciones Java, ayudando al desarrollo de proyectos que utilicen todas las tecnologías Java: Java SE, Java EE, embebidas y en la nube. El IDE es modular y se puede ampliar a través de complementos. Se puede ejecutar en sistemas operativos en los que el kit de desarrollo de java (JDK) ha sido instalado (Wielenga, 2015).

NetBeans 8.0.1 es capaz de permitir un construir un proyecto HTML5 como una aplicación de Apache Cordova y desplegarla a dispositivo o simuladores Android o iOS. El IDE también soporta la depuración en el dispositivo o en simulador móvil. Para eso se necesita instalar Córdoba en el sistema local para empaquetar una aplicación HTML5 como una aplicación móvil nativa con NetBeans IDE. Se debe instalar el gestor de paquetes NodeJS, para instalar y actualizar Córdoba. También tendrá que confirmar que Git está instalado en el sistema local y configurado correctamente. Cordova usa Git para recuperar los archivos de origen Córdoba, necesarios desde un repositorio al empaquetar la aplicación como una aplicación móvil nativa (Netbeans Org, 2014)

Apache Tomcat es un contenedor de Servlets y de web basada en java de código abierto, que se utiliza para alojar aplicaciones basadas en Java. Fue desarrollado por primera vez para Jakarta Tomcat. Debido a un aumento en la demanda, más tarde se alojó como un proyecto separado llamado Apache Tomcat, que es respaldado por The Apache Software Foundation. Fue inicialmente desarrollado por James Duncan Davidson, un arquitecto de software en Sun Microsystems. Más tarde, ayudó a que este proyecto fuera de código abierto y jugó un papel clave en la donación de este proyecto de Sun Microsystems a The Apache Software Foundation. Tomcat implementa las especificaciones **Java Servlet** y **JavaServer Pages (JSP)** de Sun Microsystems, y proporciona un entorno de servidor web HTTP "puro Java" para que se ejecute el código de Java (Khare, 2012).

En la siguiente figura se muestra las relaciones de uso, entre las aplicaciones que forman parte de la plataforma software para el funcionamiento de apache cordova (que será utilizado dentro del ide netbenas para poder desarrollar la aplicación móvil.



**Figura 5.26 Relaciones de uso entre los componentes de la plataforma de desarrollo del sistema (Fuente: Elaboración propia)**

## Hardware

Para el desarrollo de este trabajo de investigación se ha utilizado el siguiente hardware:

### ❖ Laptop con las siguientes características:

Procesador: Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz

Memoria instalada (RAM): 12Gb

Sistema operativo Windows 8 de 64 bits, procesador x 64

Disco duro: 500 Gb

### ❖ Smartphone con las siguientes características:

Smsung Galaxy S4

Modelo: GT-I9500

Versión de sistema operativo: Android 4.4.2

### ❖ Router

Cisco Model DPQ3925 8x4 DOCSIS 3.0

## 5.2.2 Frameworks y web services

### Implementados en el Componente web

#### JSF 2.2, Primefaces 4.0, Start 1.0.10

Según la especificación de JSF 2.3 (Oracle, 2017a), Java server faces es un framework de interfaz de usuario (UI) para aplicaciones web de java. Es diseñado para aliviar

significativamente la carga de escribir aplicaciones mantenibles que se ejecuten en un servidor de aplicaciones java y renderizar sus interfaces de usuarios en clientes de destino. (Oracle, 2017a) detalla las formas en que este framework ayuda al desarrollo de las aplicaciones:

- Hace fácil la construcción de interfaces de usuario desde el conjunto de componentes de interfaces de usuario reusables.
- Simplifica la migración de datos de aplicaciones para y desde la interfaz de usuario.
- Ayuda a gestionar el estado de las interfaces de usuario a través de las peticiones al servidor.
- Provee un modelo simple para escribir eventos generados desde el cliente al código de la aplicación del lado del servidor.
- Permite construir y reusar componentes de interfaz de usuario fácilmente.

Por otro lado, de acuerdo a (Jonna, 2014), PrimeFaces es un conjunto de componentes ligeros basados en Java Server Faces. Provee un conjunto de más de 100 componentes de interfaz de usuario con único jar que no requiere configuraciones y dependencias. De acuerdo a (Jonna, 2014), PrimeFaces provee de las siguientes características:

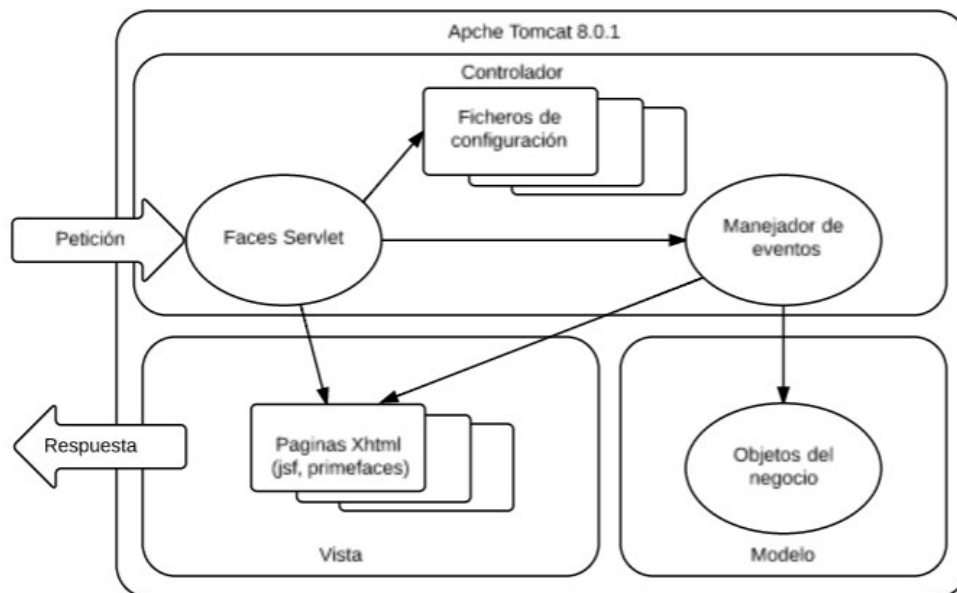
- Más de 20 componentes de interfaz de usuario
- Soporte para componentes de utilidad comunes, comportamientos de clientes, conversores y herramientas de optimización, como el plugin optimizador de recursos Maven
- Compatibilidad con Ajax incorporada utilizando widgets jQuery
- Configuración cero y no hay dependencias de biblioteca de terceros para componentes mayoritarios
- Soporte temático para todos los componentes
- Compatibilidad con varios servidores (IE8 +, Chrome, Firefox, Safari y Opera)

La figura a continuación muestra la configuración del archivo web.xml, realizada para habilitar el framework Jsf y elegir el tema “start” que se ha elegido para este trabajo.

<pre> &lt;servlet&gt;   &lt;servlet-name&gt;Faces Servlet&lt;/servlet-name&gt;   &lt;servlet-class&gt;javax.faces.webapp.FacesServlet&lt;/servlet-class&gt;   &lt;load-on-startup&gt;1&lt;/load-on-startup&gt; &lt;/servlet&gt; &lt;servlet-mapping&gt;   &lt;servlet-name&gt;Faces Servlet&lt;/servlet-name&gt;   &lt;url-pattern&gt;*.xhtml&lt;/url-pattern&gt; &lt;/servlet-mapping&gt; &lt;session-config&gt;   &lt;session-timeout&gt;     30   &lt;/session-timeout&gt; &lt;/session-config&gt; &lt;welcome-file-list&gt;   &lt;welcome-file&gt;vista/login.xhtml&lt;/welcome-file&gt; &lt;/welcome-file-list&gt; &lt;context-param&gt;   &lt;param-name&gt;primefaces.THEME&lt;/param-name&gt;   &lt;param-value&gt;start&lt;/param-value&gt; &lt;/context-param&gt; </pre>	<p>Para que las peticiones sean dirigidas al servlet FacesServlet</p> <p>Se define el patrón que va admitir el Servlet FacesServlet</p> <p>Especifica el tiempo de espera en minutos, antes de cerrar la sesión automáticamente</p> <p>Se define la archivo que será mostrado al correr el programa</p> <p>Se elige el tema que nos brinda primefaces</p>
--	---

**Figura 5.27 Configuración seguida para la habilitación del framework Jsf y del tema Start de primefaces (Fuente: Elaboración propia)**

En la figura siguiente figura de acuerdo a (Ceballos, 2008), se muestra una vista la arquitectura del framework Jsf, donde se ha incluido además al papel de Primefaces.



**Figura 5.28 Arquitectura del framework Jsf con Primefaces (Fuente: Adaptado de: (Ceballos, 2008))**

### Hibernate 4.3

Según (Minter, Linwood, & Ottinger, 2010): “Hibernate es uno de los frameworks ORM más ampliamente usados en la industria. Provee todos los beneficios de una solución ORM e implementa la especificación de JPA”.

Los principales componentes de Hiberante son los siguientes:

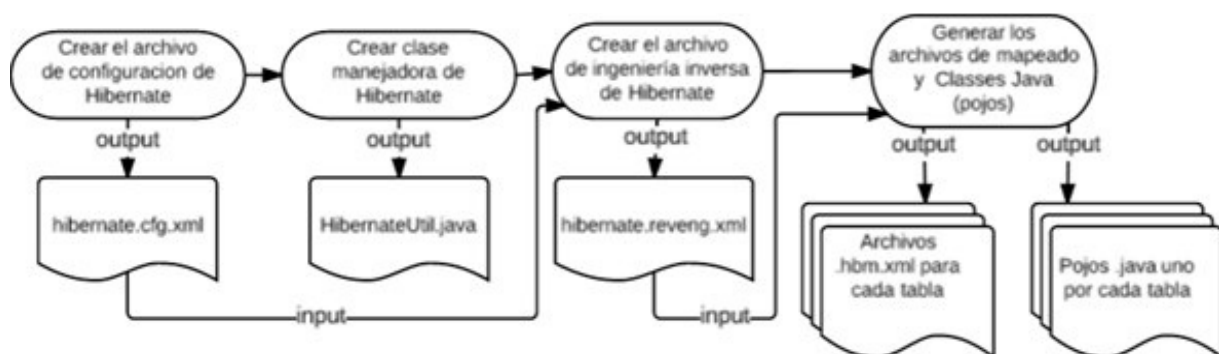
**Hibernate Core:** El núcleo genera SQL y alivia el manejo manual del conjunto de resultados de la Java Database Connectivity (JDBC) y las conversiones de objetos. Metadatos son definidos en archivos XML simples. El núcleo ofrece varias opciones para

escribir consultas: SQL plano; Hibernate Query Language (HQL), que es especificado para Hibernate; criterios programáticos y Query by Example (QBE). Se puede optimizar carga de objetos con varias opciones de recuperación (fetching) almacenamiento en cache (caching).

**Hibernate Annotations:** Hibernate provee la opción de definir metadatos usando anotaciones. Esta característica reduce la configuración usando archivos XML y lo hace fácil de definir los metadatos requeridos directamente en código fuente de Java.

**Hibernate EntityManager:** La especificación JPA define interfaces de programación, reglas de ciclo de vida para objetos persistentes, y funciones de consultas. (Minter et al., 2010).

Netbeans ofrece una herramienta para la generación de código de los archivos de mapeado y de los Pojos (o clases mapeadas a Java equivalentes a cada tabla de la base de datos) utilizando la ingeniería inversa. En la figura siguiente se muestra un diagrama donde se representan a las actividades que se realizaron para generar los archivos necesarios para habilitar el framework Hibernate dentro del proyecto.



**Figura 5.29 Actividades realizadas con el ide netbeans para la generación de los archivos necesarios para el manejo del framework Hibernate (Fuente: Elaboración propia)**

### Crear el archivo de configuración de Hibernate

El archivo de configuración almacena información de la conexión a la base de datos. A través de netbenas se puede editar el archivo con el editor de múltiples vistas, o editar el XML directamente en el editor XML.

### Crear clase manejadora de Hibernate

Se creaa una clase de ayuda para el manejo del arranque y que usa la clase `SessionFactory` para obtener un objeto de la interfaz `Session`. La clase invoca a el método `configure()`, carga la configuración en `hibernate.cfg.xml` y luego construye el `SessionFactory` para conseguir una instancia de `Session` con todos los parámetros establecidos dentro del archivo de configuración

del hibernate, como son: usuario de base de datos, password, y otros parámetros que se utilizarán para el funcionamiento de Hibernate .

### Crear el archivo de ingeniería inversa de Hibernate

Para utilizar los archivos de Mapeo Hibernate y los POJOs desde el asistente de base de datos, primero se debe crear el archivo de ingeniería inversa hibernate.reveng.xml.

El archivo de la ingeniería inversa permite tener un mayor control sobre la estrategia de asignación de base de datos. El Asistente para ingeniería inversa Hibernate crea un archivo de ingeniería inversa con una configuración por defecto que se puede editar en el editor XML.

### Generar los archivos de mapeado y Clases Java (POJOs)

Con los archivos de onfiguración y de ingeniería inversa creados, el asistente de base de datos de Netbeans permite realizar la generación de los xml de mapeos y de los POJOs desde la base de datos. En este proceso, se generará un xml y una clase java por cada tabla. Los XML contienen las columnas de las tablas y que se asignan a los campos en los POJOs, relaciones de multiplicidad entre tablas, etc.

En la figura siguiente se muestra a los archivos del framework hibernate y como se relaciona con el resto del sistema.

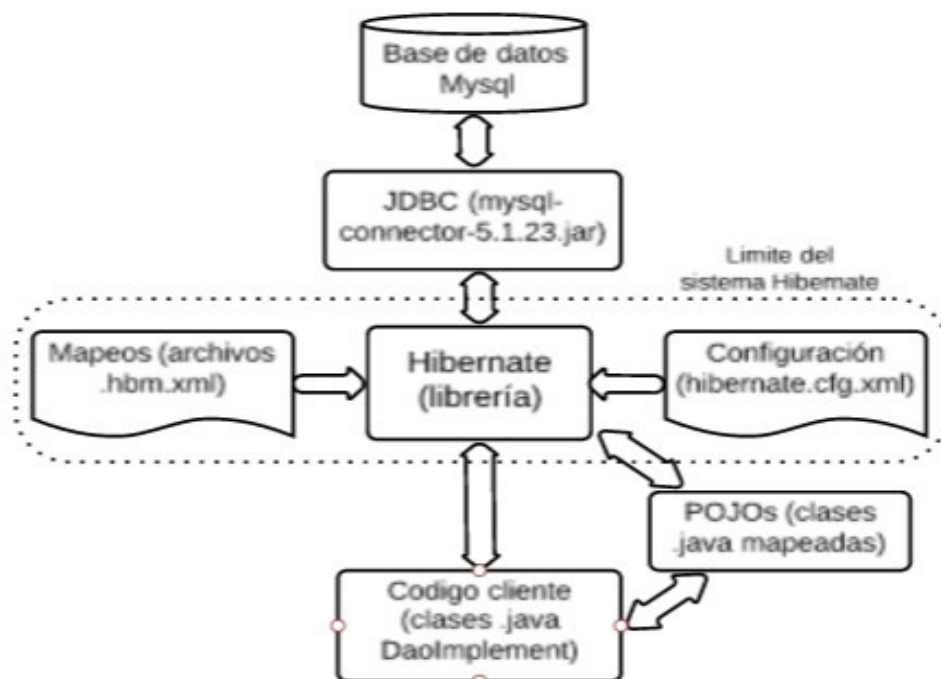


Figura 5.30 El rol de Hibernate en el componente web (Fuente: Adaptado de (Minter et al., 2010))

### Web Services

JAX-RS 1.1, Jersey 1.3, JAX-B 2.2.5 y Json

JAX-RS:

“Esta especificación define un conjunto de APIs de Java para el desarrollo de servicios web contruidos de acuerdo con el estilo arquitectónico Representational State Transfer (REST)” (Oracle, 2017b).

Básicamente en JAX-RS se puede describir sus métodos que sirven para manejar las peticiones del cliente, por ello, en la tabla siguiente se muestra el conjunto de solicitudes que pueden ser realizadas por el cliente a través de http, y sus correspondientes anotaciones para los métodos del lado del servidor.

<b>Método http</b>	<b>Anotación para los métodos de los web service</b>
GET	@GET
POST	@POST
PUT	@PUT
DELETE	@DELETE
HEAD	@HEAD
OPTIONS	@OPTIONS

**Tabla 5.10 Anotaciones JAX-RS (Fuente: Adaptado de (Oracle, 2017b))**

Los métodos de recursos pueden tener un conjunto de parámetros. Algunos de estos parámetros tendrán anotaciones y otros no. Se puede anotar los parámetros del método con una de las anotaciones en siguiente tabla:

<b>Método http</b>	<b>Parámetros para los métodos del web services</b>
@MatrixParam	Extrae el valor de parámetro de la matriz URI
@QueryParam	Extrae el valor de parámetro de la consulta
@PathParam	Extrae el valor del parámetro de la plantilla de la URI
@CookieParam	Extrae el valor de la cookie
@HeaderParam	Extrae el valor de la cabecera
@FormParam	Extrae el valor de un formulario
@Context	Inyecta una instancia de un recurso soportado

**Tabla 5.11 Parámetros proveídos por el api JAX-RS para el uso de web services (Fuente: Adaptado de Oracle, 2017b))**

### **Jersey:**

De acuerdo a (Gulabani, 2014): “ Jersey [...] es la implementación de referencia de JAX-RS [...]”.

### **JAX-B y Json**

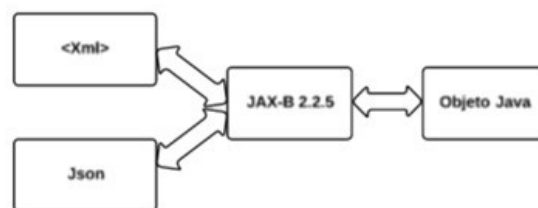
De acuerdo a (Oracle, 2014), la arquitectura Java para “binding” de XML (JAXB) es una tecnología de enlace XML-a-Java que simplifica el desarrollo de servicios web al permitir transformaciones entre esquemas y objetos Java y entre documentos de instancia XML e instancias de objetos Java. Un esquema XML define los elementos de datos y la estructura de un documento XML. Se puede usar las APIs y las herramientas JAXB para establecer asignaciones entre las clases Java y los esquemas XML. La tecnología JAXB proporciona las herramientas que le permiten convertir los documentos XML a y desde objetos Java.

(Oracle, 2014) también nos habla sobre la utilización de JAX-RS con JAXB: “JAX-RS puede leer y escribir automáticamente XML utilizando JAXB, pero también puede funcionar con datos JSON. JSON es un formato simple basado en texto para el intercambio de datos derivado de JavaScript”.

Por ejemplo, un ejemplo de la representación Json de un producto:

```
{  
  "id": "1",  
  "nombre": "MaxiQueso",  
  "descripción": "Queso fresco natural",  
  "price": 32  
}
```

En la siguiente figura se muestra las diversas conversiones que puede realizar JAX-B.



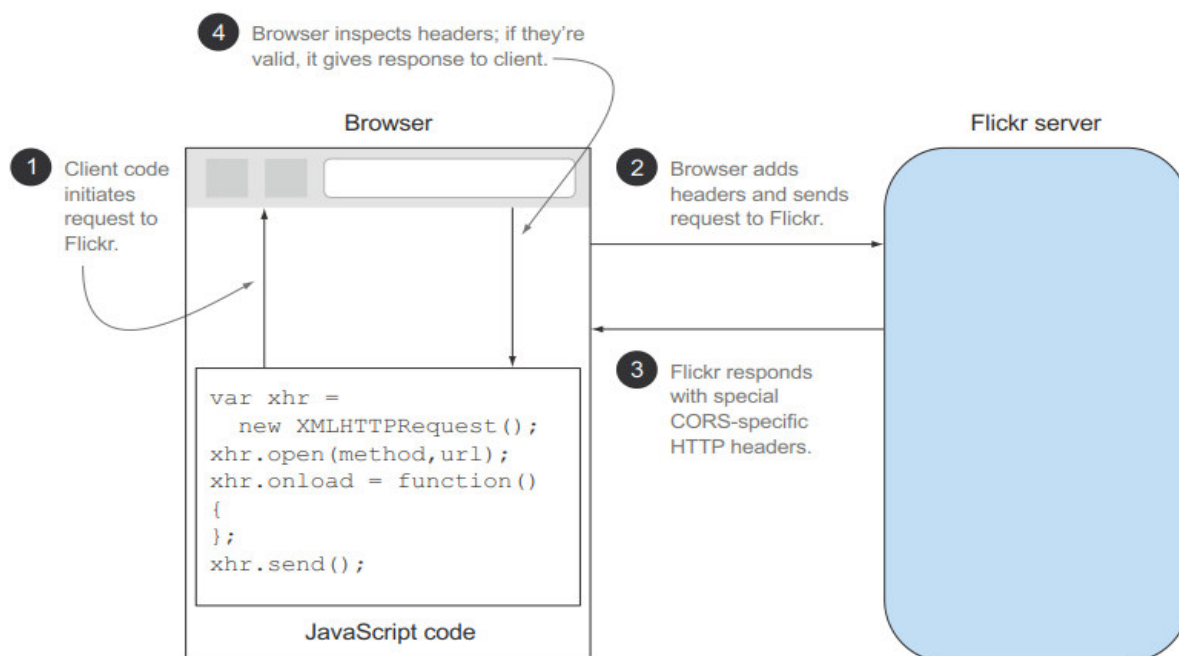
**Figura 5.31 Conversiones hechas por JAX-B (Fuente: Elaboración propia)**

## Cors

De acuerdo a (Hossain, 2015), es una tecnología web que tiene un componente del lado del servidor y un componente del lado cliente que permite a los clientes web realizar solicitudes HTTP a servidores alojados en diferentes orígenes.

En la siguiente figura se muestra un ejemplo de los CORS en acción:





**Figura 5.32 Flujo de petición de extremo a extremo de los CORS (Fuente: (Hossain, 2015))**

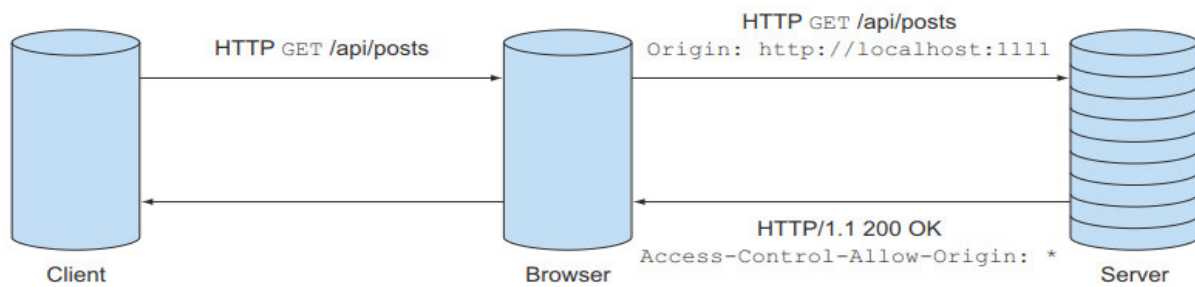
De acuerdo al ejemplo (Hossain, 2015), explica el flujo de trabajo de los CORS en los siguientes pasos:

- 1) La petición de los CORS se inicia con el código del cliente de JavaScript.
- 2) El navegador incluye encabezados HTTP adicionales en la solicitud antes de enviar la solicitud al servidor.
- 3) El servidor incluye encabezados HTTP en la respuesta que indican si la solicitud está permitida.
- 4) Si la solicitud está permitida, el navegador envía la respuesta al código del cliente.

“En el lado del componente servidor configura qué tipos de solicitudes de origen cruzado están permitidas, mientras el componente del lado del cliente controla cómo se realizan las solicitudes de origen cruzado”.

De acuerdo (Hossain, 2015), el encabezado que se utiliza por parte del cliente y del servidor para las peticiones de CORS se llama Access-Control-Allow-Origin. El valor del encabezado Access-Control-Allow-Origin puede ser un comodín (Ejm: \*) o un valor de origen (Ejm: http://localhost:8090).

En la siguiente figura se muestra un flujo exitoso de una petición hecha por el cliente y una respuesta con cabecera Access-Control-Allow-Origin: \* que indica que las solicitudes están permitidas para cualquier origen:



**Figura 5.33 Respondiendo a una solicitud CORS usando el encabezado Access-Control-Allow-Origin Fuente: (Hossain, 2015))**

Tomcat provee un número de filtros que pueden ser establecidos, para configurar la cabecera (Access-Control-Allow-Origin) que viajará en la respuesta desde el servidor al cliente. El nombre de la clase para el filtro de CORS es org.apache.catalina.filters.CorsFilter (Apache Software Foundation, 2010).

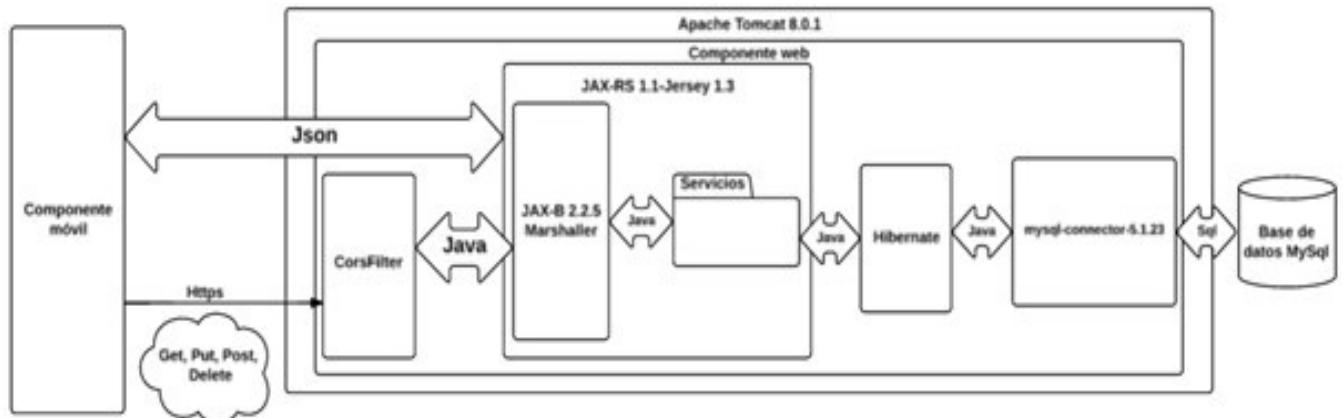
En la siguiente figura se muestran parámetros de inicialización dentro del archivo WEB-INF/web.xml para permitir el acceso de origen cruzado para todos los clientes (\*) y con cualquier método de petición GET, POST, PUT...:

La imagen muestra un editor de código con el archivo web.xml. Se han marcado dos secciones con recuadros rojos y números: 1. El primer `<init-param>` que configura `cors.allowed.origins` con el valor `*`. 2. El segundo `<init-param>` que configura `cors.allowed.methods` con el valor `GET,POST,HEAD,OPTIONS,PUT`. El archivo también incluye configuraciones para `com.sun.jersey.api.json.POJOMappingFeature`, `cors.allowed.headers` y `cors.exposed.headers`.

**Figura 5.34 Configuración de los CORS desde el lado del servidor en la aplicación para permitir peticiones de diversos orígenes (Fuente: Elaboración propia)**

En la figura siguiente se presenta un esquema que representa la comunicación entre diferentes elementos que componen la aplicación distribuida. De manera remota desde el componente móvil a través de peticiones http hacia los web services, localizados en el componente web (servidor), el flujo de los datos en formato Json se realiza para la recuperación de datos desde el servidor o para la inserción de datos hacia el servidor, la clase CorsFilter es habilitada en el archivo web.xml del componente web (servidor, y que está incluida en las librerías del servidor web Apache Tomcat), y que permite que las peticiones provenientes del cliente móvil no sean rechazadas. Los elementos como el framework Jersey es encargado de serializar los objetos en

formato Json o Xml que son convertidos desde un objeto java a través de la acción “marshall” de JAX-B o de deserializar los objetos java que han sido convertidos en objetos java mediante la acción “unmarshall” de JAX-B. El paquete de servicios, utiliza la capa de acceso a datos, que es a través de la cual se realiza las diversas peticiones a hibernate y éste mediante el JDBC para mysql recupera o insertar los datos en la base de datos.

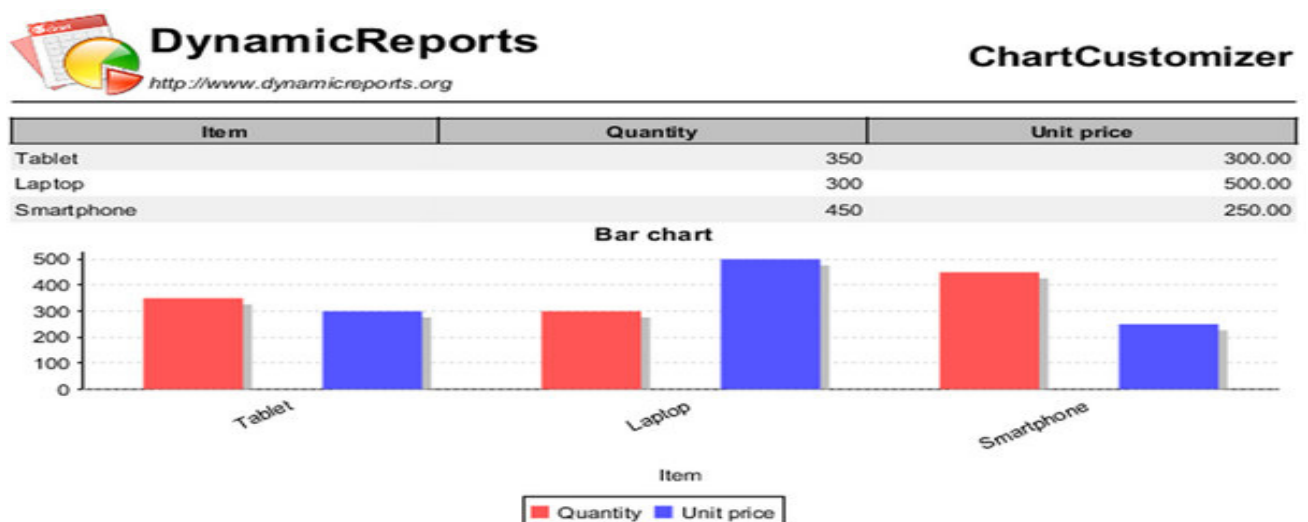


**Figura 5.35 Esquema de la comunicación de los componentes de la aplicación distribuida (Fuente: Elaboración propia)**

## DynamicReports

DynamicReports es una librería de código abierto basada en JasperReports para la generación de reportes basados en complejos conjunto de datos. DynamicReports soporta un amplio rango de formato de datos, incluyendo bases de datos relacionales, archivos XML, XLS, y CVS, entre otros (Di Nitto, Matthews, Petcu, & Solberg, 2017).

En la figura siguiente se presenta una imagen de un reporte dinámico compuesto por un encabezado formado por un logo y texto, una tabla, un cuadro estadístico y también una leyenda.



**Figura 5.36 Ejemplo de reporte dinámico hecho con DynamicReports (Fuente: (“DynamicReports”, s. f.))**

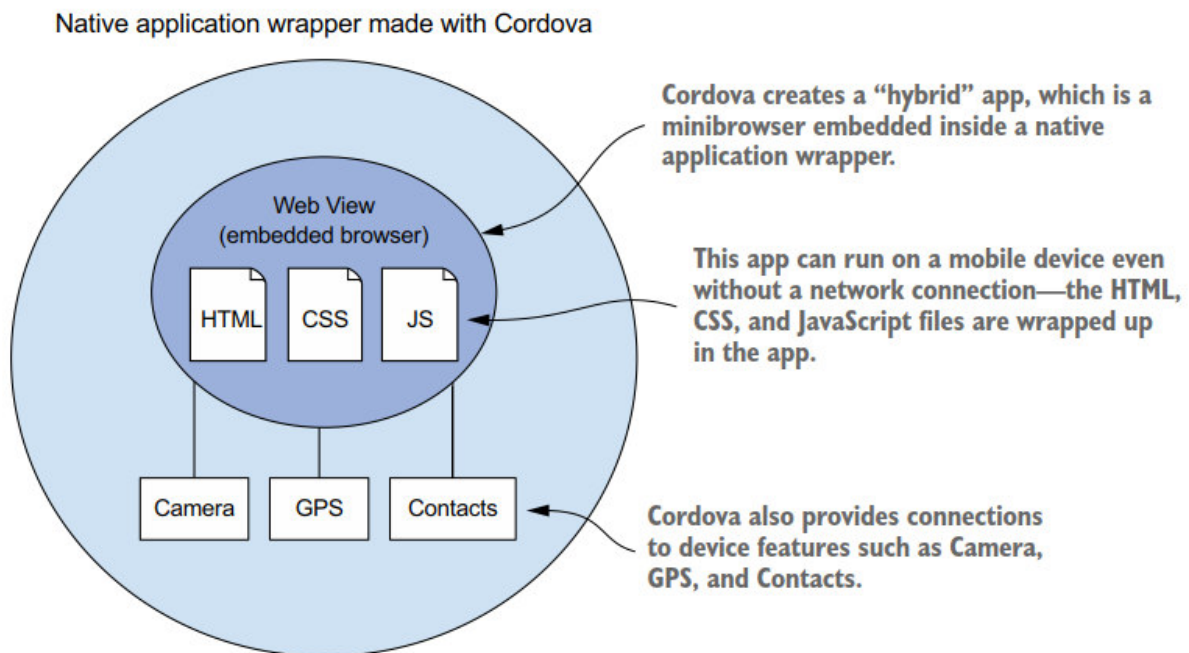
## Componente móvil

### Apache Cordova, Html5, JQuery Mobile 4.4.5

Según (R. K. Camden, 2016):

Cordova es un framework de código abierto que permite convertir HTML, JavaScript y hojas de estilo en cascada (CSS) en una aplicación nativa que se puede ejecutar en iOS, Android y otras plataformas móviles. Cordova usa un "envoltorio" nativo alrededor de una vista web (piense en ello como un navegador embebido), comúnmente llamada una aplicación móvil híbrida. También proporciona acceso a características de hardware como la cámara y el acelerómetro. A diferencia de una simple página web, las aplicaciones Cordova se pueden encontrar (¡y vender!) en las tiendas de aplicaciones al igual que las aplicaciones nativas.

La Figura siguiente ofrece una demostración visual de cómo las aplicaciones móviles híbridas trabajan.



**Figura 5.37** Una aplicación simple como una app híbrida trabaja (Fuente: (R. K. Camden, 2016))

De acuerdo a la imagen anterior, (R. K. Camden, 2016) explica el funcionamiento de Cordova: Cordova crea una aplicación híbrida que es un mini browser embebido dentro del envoltorio de la aplicación nativa.

Esta aplicación puede ejecutarse sobre un dispositivo móvil incluso sin conexión a red. Los archivos HTML, CSS y el JavaScript son envueltos en la app.

Cordova también provee conexiones a las características de los dispositivos tales como cámara, GPS y contactos. (R. K. Camden, 2016).

## **Html5**

HTML5 está destinado a ser el sucesor de HTML4 y reemplazará a XHTML. HTML5 incorpora características de HTML y XHTML, agrega nuevos elementos propios, proporciona nuevas características tales como ediciones de formularios y videos nativos, y está diseñado para ser compatible con versiones anteriores.

El W3C aprobó HTML5 para el estado de Candidate Recommendation a finales de 2012. HTML5 alcanzó el estado final de la Recomendación a fines de 2014. El W3C continúa su desarrollo de HTML y está agregando más elementos nuevos, atributos y características en HTML 5.1, que es actualmente en forma de borrador.

Las versiones recientes de navegadores populares, como Internet Explorer, Microsoft Edge, Firefox, Safari, Google Chrome y Opera ya son compatibles con la mayoría de las funciones de HTML5 (Felke-Morris, 2017).

## **Jquery Mobile 1.4.5**

De acuerdo a (R. Camden & Matthews, 2016), jQuery Mobile es un framework de interfaz de usuario que permite convertir sitios web regulares de HTML y CSS en sitios web optimizados para dispositivos móviles. Es una forma poderosa y práctica herramienta para crear sitios web móviles que cualquier desarrollador HTML existente puede recoger y adaptar. jQuery Mobile es mucho más amigable para las personas que están más familiarizadas con HTML en lugar de JavaScript. jQuery Mobile es táctil, lo que tener sentido para cualquier persona que haya usado un teléfono inteligente y haya tenido problemas para hacer clic en el lugar en un sitio web con texto pequeño y enlaces difíciles de detectar.

### **5.2.3 Implantación de la solución**

#### ***Descripción del caso de estudio***

La fase de producción de este sistema se realizó gracias a la empresa de productos lácteos Artemisa localizado en la Jr. El Corregidor 111 en la Urb. Latina del distrito de José Leonardo Ortiz en la ciudad de Chiclayo en el departamento de Lambayeque. Debemos realizar el estudio de los resultados obtenidos mediante el uso de esta aplicación que se ha desarrollado para poder validar los objetivos establecidos en este trabajo.



**Figura 5.38 Presentación de Artemisa Lácteos (Fuente: (Artemisa, 2018))**

En la figura siguiente se muestran los principales productos de Artemisa:



**Figura 5.39 Algunos productos de Artemisa Lácteos (Fuente: (Artemisa, 2018))**



Los escenarios que se llevan a cabo son en función a los principales casos de uso del sistema que se han construido en la fase de implementación para hacer frente a la problemática descrita.

#### A) Registro de pedidos:

Necesitamos poner a prueba el tiempo en los registros de pedidos. Syslac soporta el registro de pedidos tanto desde su componente web como su componente móvil. Para eso se realizaron los siguientes escenarios:

- \* Registro de pedidos de forma manual sin Syslac.
- \* Registro de pedidos con el aplicativo web de Syslac.
- \* Registro de pedido con el aplicativo móvil de Syslac.

#### B) Facturación de pedidos de uno en uno:

La facturación de pedidos se realiza mediante el módulo de facturación de pedidos del componente web de Syslac. Se necesita evaluar el tiempo de facturación de los pedidos. Para lo cual se realizarán los siguientes escenarios:

- \* Facturación de pedidos de uno en uno sin el uso de Syslac.
- \* Facturación de pedidos de uno en uno usando Syslac.

#### C) Facturación de pedidos de forma masiva:

La facturación de pedidos de forma masiva se realiza mediante el componente web de Syslac. La facturación manual no se puede realizar de forma masiva por lo que solo se considera solo el siguiente escenario:

- \* Facturación de forma masiva usando el módulo de facturación de Syslac.

#### D) Emisión de reportes.

- \* Emisión de reportes para las ventas por producto en los últimos meses.
- \* Emisión de reportes para las ventas por cliente en los últimos meses.

### ***Detalles de la implantación y estadísticas del sistema***

Las entradas necesarias para los casos establecidos vienen a ser la información de los pedidos que son realizados por los clientes. Para realizar este análisis se han tomado los pedidos realizados por 20 clientes frecuentes de esta empresa como muestra:

Pedidos de personas naturales:

Pedido	Detalles del pedido	Delivery
1	2 yogures de fresa, 2 yogures de durazno, 2 kilos de queso.	No
2	3 yogures de frutos secos, 3 yogures de guanábana, 1 kilo de queso Artemisa.	No
3	3 yogures de frutos secos, 2 yogures de fresa.	No
4	3 yogures de frutos secos, 2 kilos de queso light.	No
5	3 yogures de frutos secos, 3 yogures de mango y 2 de fresa.	No
6	2 yogures de fresa, 2 yogures de durazno, 1 kilos de queso.	Si
7	3 yogures de frutos secos, 3 yogures de guanábana.	Si
8	4 yogures de frutos secos, 2 kilos de queso.	Si
9	3 yogures de frutos secos, 1 kilos de queso.	Si
10	1 yogur de frutos secos, 3 yogures de mango y 1 de fresa.	Si

**Tabla 5.12 Pedidos de personas naturales que se tomaron de muestra de la implantación del sistema (Fuente: Elaboración propia)**

Pedidos de personas jurídicas

Pedido	Detalles del pedido	Delivery
1	7 yogures de fresa, 5 yogures de durazno, 4 kilos de queso light, 2 kilos de mantequilla.	No
2	8 yogures de frutos secos, 10 kilos de queso mozzarella.	No
3	8 yogures de fresa, 6 yogures de durazno, 6 yogures de frutos secos, 5 Kg. de queso Artemisa.	No
4	8 yogures de fresa, 5 yogures de durazno, 5 Kg de queso Artemisa.	No
5	6 yogures de lucuma, 6 yogures de frutos secos, 6 yogures de mango, 4 kilos de queso Artemisa.	No
6	13 kilos de queso mozzarella.	Si
7	6 yogures de fresa, 8 kilos de queso Atemisa.	Si
8	7 yogures de fresa, 2 yogures de durazno, 2 kilos de queso Artemisa.	Si
9	8 yogures de fresa, 3 kilos de queso Artemisa.	Si
10	4 yogures de fresa, 4 yogures naturales, 6 kilos de queso Artemisa.	Si

**Tabla 5.13 Pedidos de personas jurídicas que se tomaron como muestra de la implantación del sistema (Fuente: Elaboración propia)**



Los resultados de los escenarios establecidos previamente se detallan a continuación:

#### A) Tiempos del Registro de pedidos de la muestra

Los tiempos de la toma de pedido sin Syslac contemplan el registro de los datos del cliente, búsqueda del precio de los productos para el grupo de precios al que pertenece el cliente, el cálculo del precio total del pedido y el tiempo de llenado manual en el libro de registro de pedidos.

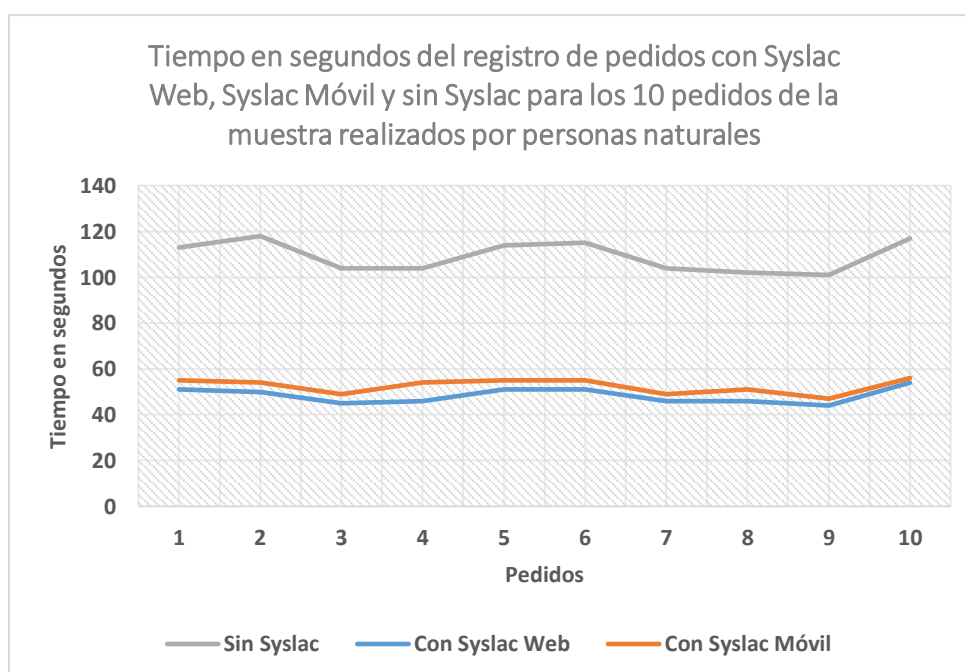
Dentro de estos tiempos se consideró el tiempo de interacción del vendedor con el sistema (búsqueda, selección de productos con los precios correspondientes al grupo de precios al que pertenece el cliente, el ingreso de las cantidades solicitadas al sistema) para concretar el pedido a través de Syslac Web y Syslac Móvil.

En la siguiente tabla se presentan los tiempos del registro de pedidos sin Syslac, con Syslac Web y con Syslac Móvil para los 10 pedidos de la muestra realizados por personas naturales:

Escenario	Pedidos									
	1	2	3	4	5	6	7	8	9	10
Sin Syslac	113	118	104	104	114	115	104	102	101	117
Con Syslac Web	51	50	45	46	51	51	46	46	44	54
Con Syslac Móvil	55	54	49	54	55	55	49	51	47	56

**Tabla 5.14 Tiempo en segundos del registro de pedidos con Syslac Web, con Syslac móvil y Sin Syslac para los 10 pedidos de la muestra realizados por personas naturales (Fuente: Elaboración propia)**

En la siguiente figura se muestran los tiempos del registro de pedidos sin Syslac, con Syslac Web y con Syslac Móvil para las 10 personas naturales de la muestra:



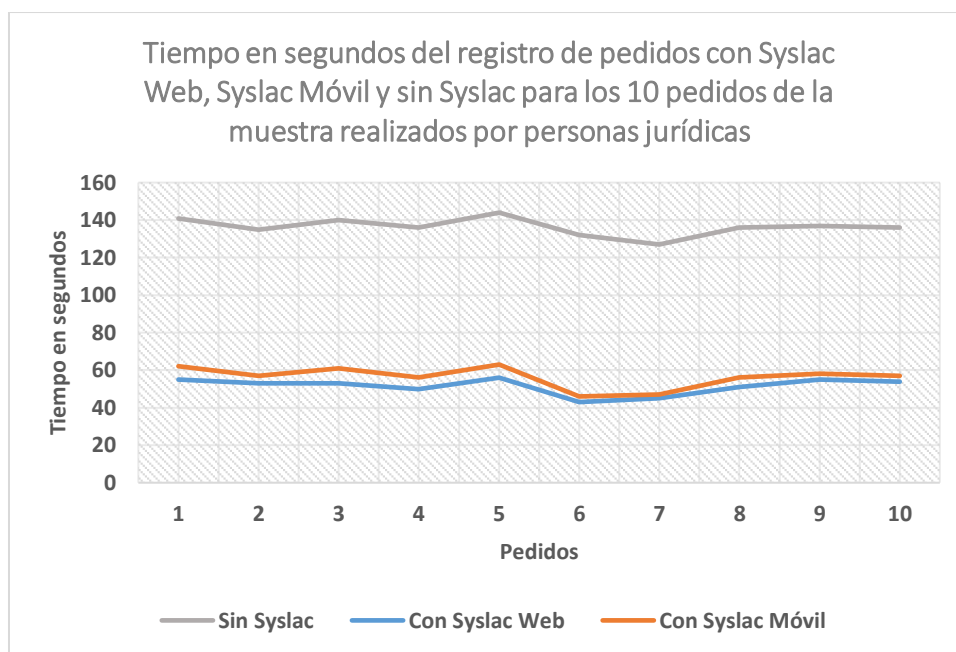
**Figura 5.40 Gráfico de tiempo en segundos del registro de pedidos con Syslac Web, Syslac Móvil y sin Syslac para los 10 pedidos de la muestra realizados por personas naturales (Fuente: Elaboración propia)**

En la siguiente tabla se muestran los tiempos en segundos del registro de pedidos sin Syslac, con Syslac Web y con Syslac Móvil para los 10 pedidos de la muestra realizados por personas jurídicas:

Escenario	Pedidos									
	1	2	3	4	5	6	7	8	9	10
Sin Syslac	141	135	140	136	144	132	127	136	137	136
Con Syslac Web	55	53	53	50	56	43	45	51	55	54
Con Syslac Móvil	62	57	61	56	63	46	47	56	58	57

**Tabla 5.15 Tiempo en segundos del registro de pedidos sin Syslac, con Syslac web y con Syslac móvil para las 10 pedidos de la muestra realizado por personas jurídicas (Fuente: Elaboración propia)**

En la siguiente figura se muestran los tiempos en segundos del registro de pedidos sin Syslac, con Syslac Web y con Syslac Móvil para los 10 pedidos de la muestra realizados por personas jurídicas:



**Figura 5.41 Gráfico de tiempo en segundos del registro de pedidos con Syslac Web, Syslac Móvil y sin Syslac para los 10 pedidos de la muestra realizados por personas jurídicas (Fuente: Elaboración propia)**

B) Tiempos de facturación de pedidos de uno en uno para los pedidos de la muestra

Para la facturación uno a uno sin Syslac se consideró el tiempo de llenado manual de los formatos impresos de boletas, facturas y/o guías de remisión.

Para la facturación uno a uno con Syslac se consideró el tiempo de la asignación de cantidades a despachar luego de registrar el peso exacto de los productos a despachar, el propio proceso de facturación del sistema y la impresión de las boletas y/o facturas.

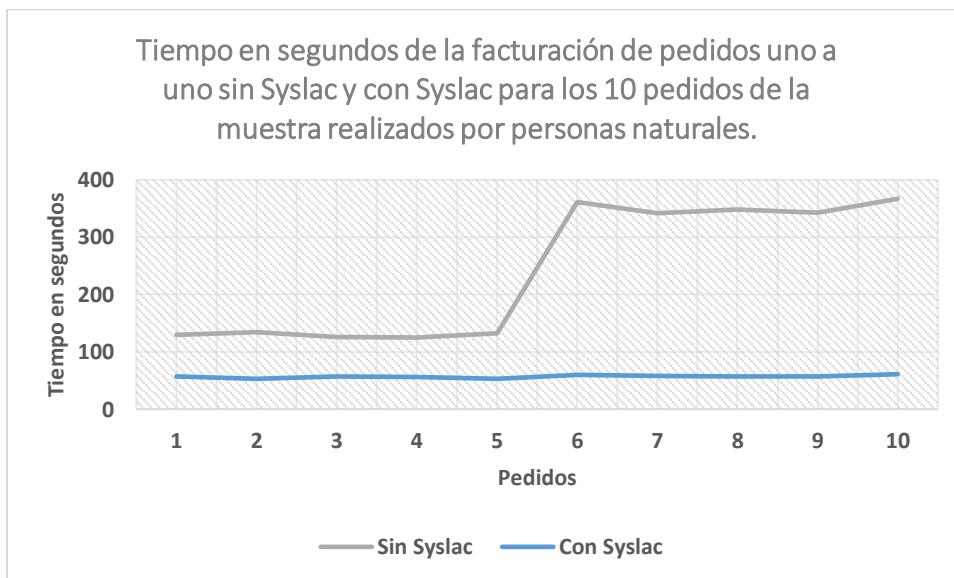
En la tabla 5.16 se muestran los tiempos en segundos de la facturación uno a uno con Syslac y Sin Syslac para los 10 pedidos de la muestra realizados por personas naturales. El tiempo total empleado para la

facturación del pedido es igual a la suma de los tiempos empleados en la emisión de la boleta y el de la emisión de guía de remisión, donde “0” significa que no se ha emitido el comprobante (para el caso de los pedidos que no son de delivery):

Escenario	Pedidos																			
	1		2		3		4		5		6		7		8		9		10	
	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
Sin Syslac	130	0	134	0	126	0	125	0	132	0	131	230	122	220	124	224	121	222	135	232
Tiempo Total	130		134		126		125		132		361		342		348		343		367	
Con Syslac	57	0	53	0	57	0	56	0	53	0	33	27	34	24	37	20	34	23	38	23
Tiempo Total	57		53		57		56		53		60		58		57		57		61	

**Tabla 5.16 Tiempo en segundos de la facturación de pedidos uno a uno sin Syslac y con Syslac para los 10 pedidos de la muestra realizados por personas naturales. Donde B, F y G significan boleta, factura y guía de remisión respectivamente (Fuente: Elaboración propia)**

En la siguiente figura se muestran los tiempos en segundos de la facturación de pedidos uno a uno sin Syslac, con Syslac Web y con Syslac Móvil para los 10 pedidos de la muestra realizados por personas naturales:



**Figura 5.42 Gráfico de tiempo en segundos de la facturación de pedidos uno a uno sin Syslac y con Syslac para los 10 pedidos de la muestra realizado por personas naturales (Fuente: Elaboración propia)**

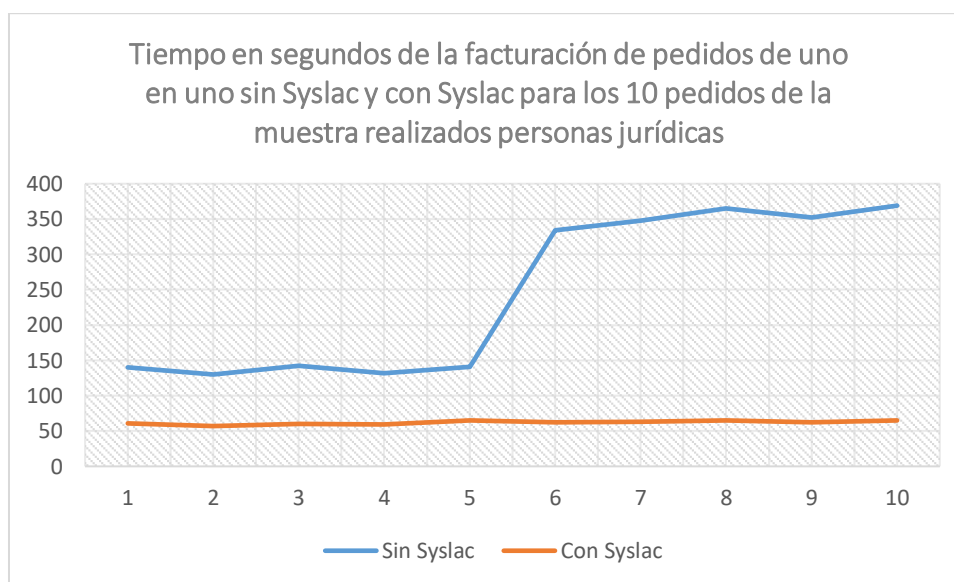
En la siguiente tabla se muestran los tiempos de los registros de pedidos de personas jurídicas sin Syslac y con Syslac para los 10 pedidos de la muestra realizados por personas jurídicas. El tiempo total empleado para la facturación del pedido es igual a la suma de los tiempos empleados en la emisión de

la factura y el de la emisión de guía de remisión, donde “0” significa que no se ha emitido el comprobante (para el caso de los pedidos que no son de delivery):

Escenario	Pedidos																			
	1		2		3		4		5		6		7		8		9		10	
	F	G	F	G	F	G	F	G	F	G	F	G	F	G	F	G	F	G	F	G
Sin Syslac	140	0	130	0	142	0	132	0	141	0	122	212	129	219	138	227	130	222	140	229
Tiempo Total	140		130		142		132		141		334		348		365		352		369	
Con Syslac	61	0	57	0	60	0	59	0	65	0	31	31	31	32	32	33	31	31	33	32
Tiempo Total	61		57		60		59		65		62		63		65		62		65	

**Tabla 5.17 Tiempo en segundos de la facturación de pedidos uno a uno sin Syslac y con Syslac para los 10 pedidos de la muestra realizados por personas jurídicas. Donde B, F y G significan boleto, factura y guía de remisión respectivamente (Fuente: Elaboración propia)**

En la siguiente figura se muestran los tiempos en segundos de la facturación de pedidos uno a uno sin Syslac y con Syslac para los 10 pedidos de la muestra realizados por personas jurídicas:



**Figura 5.43 Gráfico de tiempo en segundos de la facturación de pedidos de uno en uno sin Syslac y con Syslac para los 10 pedidos de la muestra realizados personas jurídicas (Fuente: Elaboración propia)**

#### C) Tiempos de facturación de pedidos de forma masiva para los pedidos de la muestra

Para la facturación masiva de los pedidos se consideró el tiempo de asignación de cantidades exactas de los productos a despachar, el tiempo de generación de comprobantes y de impresión de cada uno de ellos en el sistema.

En la siguiente tabla se muestran los tiempos en segundos para la facturación masiva de pedidos sin Syslac y con Syslac. El tiempo total de facturación masiva para el escenario sin Syslac no existe (NE):

Escenario	Total
Sin Syslac	NE
Con Syslac	459

**Tabla 5.18 Tiempo en segundos para la facturación de forma masiva sin Syslac y con Syslac para los 20 pedidos de la muestra (Fuente: Elaboración propia)**

D) Emisión de reportes.

La emisión de reportes para las ventas por producto en los últimos meses. El análisis de resultados de este caso será analizado por medio de la emisión de reportes para las ventas por producto en los últimos 4 meses.

Emisión de reportes para las ventas por cliente en los últimos meses. El análisis de resultados para este caso será realizado por medio de la emisión de reportes un determinado cliente y en los últimos 4 meses.

### ***Análisis de los resultados***

Para tener un mejor vistazo de los resultados obtenidos, se obtendrá las estadísticas de los tiempos del registro y de facturación de pedidos uno a uno para los pedidos de la muestra y luego se realizará el análisis respectivo para cada uno.

A) Análisis de resultados para el Registro de pedidos

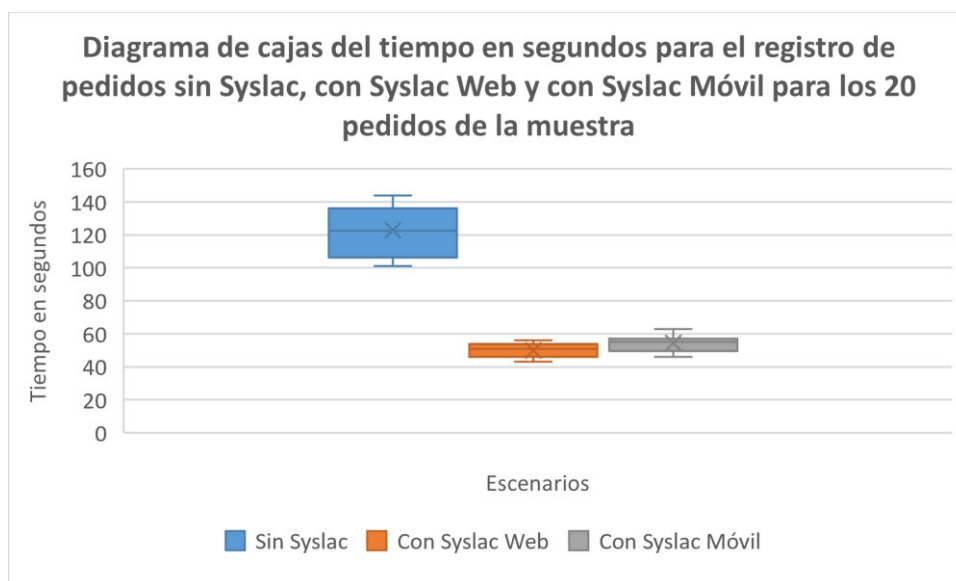
En la siguiente tabla se muestran las estadísticas de los tiempos en segundos del registro sin Syslac y con Syslac para pedidos de la muestra hechos por personas naturales y personas jurídicas (de las tablas 5.14 y 5.15):

	<i>Promedio</i>	<i>Desviación estándar</i>	<i>P25</i>	<i>P50</i>	<i>P75</i>	<i>Mediana</i>
<i>Sin Syslac</i>	122.8	15.06	110.75	122.5	136	122.5
<i>Con Syslac Web</i>	49.95	4.12	46	51	53.25	51
<i>Con Syslac Móvil</i>	54.4	4.92	50.5	55	57	55

**Tabla 5.19 Estadísticas de los tiempos en segundos del registro sin Syslac, con Syslac web y con Syslac Móvil de los 20 pedidos de la muestra (Fuente: Elaboración propia)**

Como se puede apreciar en la tabla anterior el tiempo promedio del registro de pedidos con Syslac Web fue de 49,95 segundos y el de Syslac Móvil de 54,4 segundos, siendo éstos un 59,32 % y 55,70 % menor que el tiempo de registro de pedidos sin Syslac (122,8 segundos) respectivamente, lo cual representa una mejora sustancial en la reducción de tiempos en el proceso de registro de pedidos. Respecto a la desviación estándar en los 3 escenarios se deduce que tanto en el escenario con Syslac Web (4,12 s) y Syslac Móvil (4,92 s) los datos se encuentran menos dispersos que en el escenario sin Syslac (15,6 s) de

lo que se deduce que hay mayor uniformidad en los tiempos en el Registro de pedidos usando el sistema. Para realizar el análisis comparativo gracias a los percentiles calculados se hace uso del siguiente diagrama de cajas:



**Figura 5.44 Diagrama de cajas del tiempo en segundos para el registro de pedidos sin Syslac, con Syslac Web y con Syslac Móvil para los 20 pedidos de la muestra (Fuente: Elaboración propia)**

Del anterior gráfico, se observa claramente que todos los registros de pedidos realizados con el sistema (Con Syslac Web o Syslac móvil) son mucho menores que los tiempos sin el uso del sistema.

#### **B) Análisis de resultados para la Facturación de pedidos de uno en uno:**

En la siguiente tabla se muestran las estadísticas de los tiempos en segundos de la facturación uno a uno sin Syslac y con Syslac de los pedidos de la muestra hechos por personas naturales y por personas jurídicas (de las tablas 5.16 y 5.17):

	<i>Promedio</i>	<i>Desviación estándar</i>	<i>P25</i>	<i>P50</i>	<i>P75</i>	<i>Mediana</i>
<i>Sin Syslac</i>	243.05	113.08	132	238	349	238
<i>Con Syslac</i>	59.4	3.63	57	59.5	62	59.5

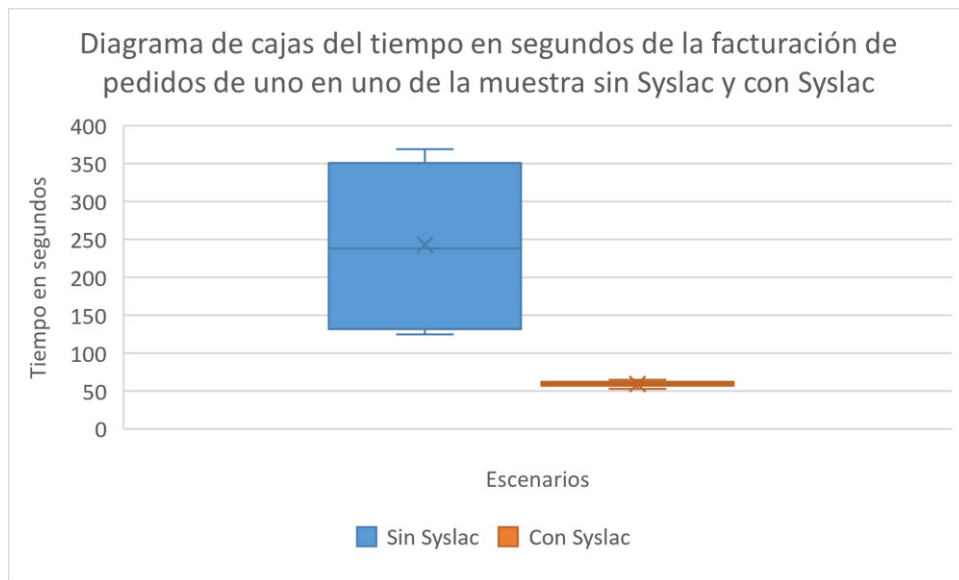
**Tabla 5.20 Estadísticas de los tiempos en segundos de la facturación uno a uno de los pedidos de la muestra para los escenarios sin Syslac y con Syslac (Fuente: Elaboración propia)**

Respecto a la tabla anterior podemos apreciar que el tiempo promedio de la facturación de pedidos de uno en uno realizado con Syslac (52,25 segundos) representa el 78,50 % menos que el tiempo de facturación promedio de uno a uno sin Syslac (243,05 segundos).

La desviación estándar que es un estadístico de dispersión que representa que tan dispersos están los datos respecto a la media; se puede ver que esta medida para la facturación de pedidos de uno en uno

con Syslac solo representa 3,63 segundos. Sin embargo, para la facturación de pedidos sin Syslac representa esta medida 113,08 segundos que es debido a que para los pedidos con “delivery” se tiene que realizar su respectiva guía de remisión, un tiempo bastante considerable, lo cual, con el uso del sistema es un tiempo adicional casi imperceptible.

Para realizar el análisis comparativo gracias a los percentiles calculados se hace uso del siguiente diagrama de cajas:



**Figura 5.45 Diagrama de cajas del tiempo en segundos de la facturación de pedidos de uno en uno de la muestra sin Syslac y con Syslac (Fuente: Elaboración propia)**

Del diagrama anterior se observa que el tiempo de facturación de uno en uno con el sistema representa una mejora bastante considerable en el tiempo de facturación.

#### C) Análisis de resultados para la Facturación de pedidos de forma masiva

De las tablas 5.16 y 5.17 obtenemos la sumatoria de la facturación manual uno a uno y la compararemos con la facturación masiva de Syslac.

Cabe recalcar que de forma manual no se puede realizar la facturación masiva de los pedidos, por lo que la implantación de Syslac realiza una mejora sustancial del proceso debido a las entradas realizadas en los casos de uso previos a la facturación para los clientes frecuentes (registro de información de los clientes, de los pedidos, de las cantidades exactas de los productos a facturar, de los productos, del transporte y de los parámetros generales).

Escenario	Facturación uno a uno	Facturación masiva
<i>Sin Syslac</i>	4861	NE
<i>Con Syslac</i>	1188	459

**Tabla 5.21 Tiempos en segundos de la facturación uno a uno vs la facturación masiva de los pedidos de la muestra sin Syslac y con Syslac (Fuente: Elaboración propia)**

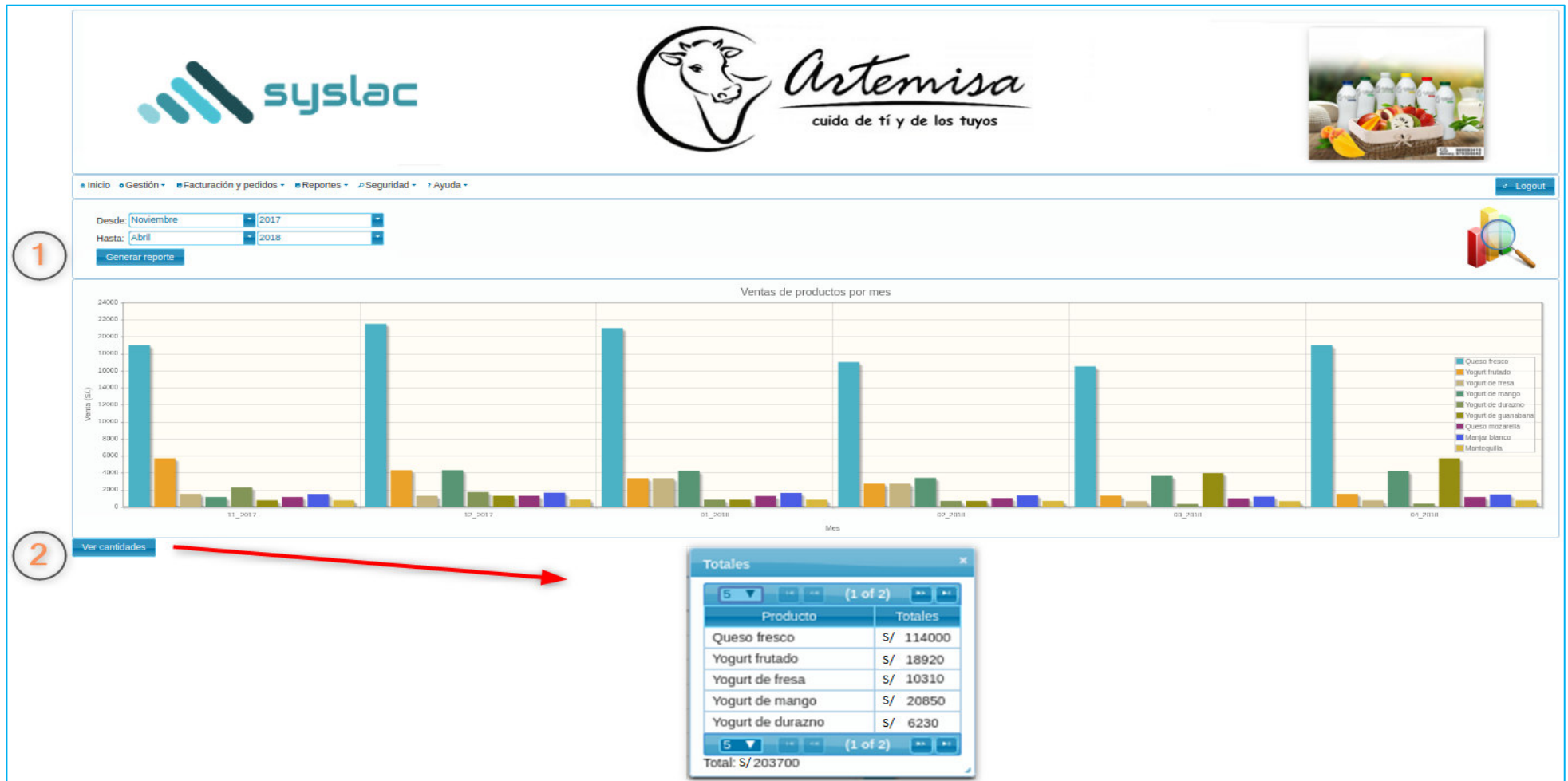
Podemos apreciar que la facturación masiva con Syslac (459 segundos) resulta el 59 % más rápida que la facturación uno a uno con Syslac, por lo que la empresa usará el modo de operación según el que mejor le convenga. También podemos apreciar que la facturación masiva con Syslac (459 segundos) representa el 90,56 % menos de tiempo que el que tomó la facturación total de los 20 pedidos de forma manual.

#### D) Emisión de reportes.

Para el caso de la emisión de reportes de las ventas por producto en los últimos meses, y de las ventas de productos por cliente en los últimos meses, a continuación, se muestra dos imágenes de dichos reportes con los datos proveídos por el área de ventas de Artemisa desde el mes de noviembre del 2017 al mes de abril del 2018:



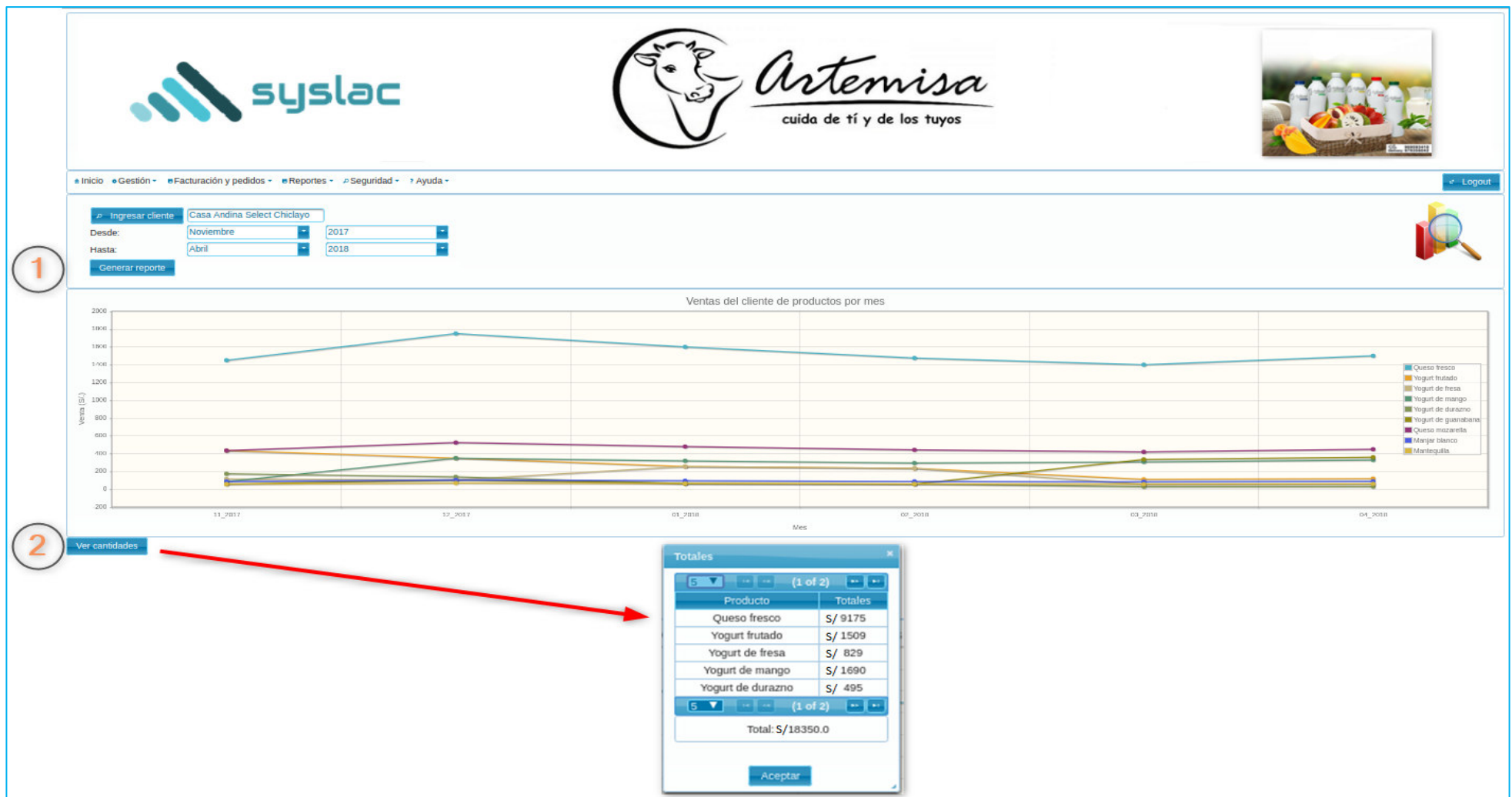
Para la emisión de reportes para las ventas por cliente en los últimos meses en la siguiente imagen se muestra el reporte de las ventas desde enero a mayo del 2018 para un cliente de la empresa.



**Figura 5.46** Reporte de ventas de los productos en los meses de noviembre del 2017 hasta abril del 2018 con los datos aproximados proveídos por el área de ventas de Artemisa (Fuente: Elaboración propia)

Como se puede apreciar el reporte de la figura 5.46 da una traza visual de las ventas por producto además de realizar un resumen de totales de los productos, lo cual ayuda a tener una mejor visión del comportamiento de las ventas.

En la siguiente figura se muestra el reporte de ventas mensuales de los productos para un determinado cliente



**Figura 5.47 Reporte de ventas de los productos en los meses de noviembre del 2017 a abril del 2018 para un cliente de la empresa, obtenido de los datos aproximados proveídos por el área de ventas de Artemisa (Fuente: Elaboración propia)**

El reporte presentado en la figura 5.47 muestra la tendencia de las ventas en el periodo señalado para un determinado cliente, lo cual ayuda a tener una visión del comportamiento de las ventas por cliente, lo que permite realizar un seguimiento de clientes más asertivo, así como poder plantear estrategias de marketing.

En este capítulo se expuso los puntos referentes al aporte práctico de la solución.

En el siguiente capítulo se expondrá las conclusiones finales de este proyecto de investigación y de los trabajos futuros que se pueden implementar en base a ésta

## CAPÍTULO VI. Conclusiones y trabajos futuros

### 6.1 Conclusiones

- Se logró desarrollar una aplicación distribuida de apoyo a la gestión de ventas para la empresa de productos lácteos Artemisa que agiliza la facturación, la captación de pedidos, la captación de clientes en campo, desde la oficina, que permite un control más seguro, que minimiza el margen de error humano, la pérdida de información, cálculos erróneos, extravío, gran almacenamiento de documentos y la mejora del servicio que brinda la empresa ayudando al planeamiento de la producción, al planteamiento de estrategias de marketing y por consiguiente a la toma de decisiones.
- Se realizó el procedimiento de selección de un estilo de arquitectónico adecuado para la aplicación distribuida a través de la técnica “benchmarking”, comparando estilos arquitectónicos de software candidatos, tomando para ello criterios de evaluación que se deriven de los requerimientos no funcionales para la aplicación distribuida, resultando seleccionado el estilo arquitectónico de software cliente servidor web.
- Se realizó el análisis y diseño del sistema del lado del cliente y del lado del servidor a través del análisis y diseño orientado a objetos (ADOO) y siguiendo la metodología RUP.
- Se seleccionó tecnologías de implementación adecuadas para la codificación de los componentes cliente y servidor de la aplicación distribuida: para el lado del cliente se seleccionó la tecnología Apache Cordova, debido a que es un framework de código abierto para la construcción de aplicaciones híbridas para móviles utilizando CSS3, HTML5, y JavaScript en vez de utilizar APIs específicas de cada plataforma móvil. Para la construcción de los webs services se seleccionó la especificación para el desarrollo de web services RESTful de java (JAX-RS). Y para la construcción del componente web se seleccionó, para la vista, las tecnologías Java Primefaces-JSF y para la persistencia de datos se utilizó el framework Hibernate y Mysql.
- Se logró implementar el componente servidor compuesto por un conjunto de web services que permiten la comunicación entre el componente cliente y el servidor, definiendo una interfaz reusable para nuevas implementaciones o adaptaciones. Se implementó la aplicación híbrida mediante el uso de las herramientas de desarrollo web de Google Chrome y de los Cors para dar permiso de acceder a los web services desde un servidor de un dominio distinto. Seguidamente esta aplicación fue empaquetada mediante el framework Apache Cordova para la plataforma móvil Android. Finalmente, para una parte de la construcción de la capa de persistencia se utilizó una técnica de ingeniería inversa el cual permite generar archivos de mapeado y clases java (Pojos) a través del framework Hibernate.

- Se realizó el análisis comparativo de los tiempos de facturación y de registro de pedidos sin y con el uso de la aplicación desarrollada mediante medidas estadísticas de resumen a partir de una muestra. De acuerdo a los resultados obtenidos y basándonos en la media como medida representativa para el análisis se obtuvo que:

El tiempo promedio de la facturación masiva con la aplicación distribuida es un 90,56 % menor que el tiempo promedio total de la facturación de forma manual.

Respecto al tiempo promedio del registro de pedidos tanto por medio del componente Web como por medio del componente Móvil es de un 59,32 % y un 55,70 % menor respecto al tiempo promedio del registro de pedidos de forma manual respectivamente.

El reporte de ventas totales de productos en los últimos meses, y el reporte de ventas de productos por cliente en los últimos meses, ofrecen una visión más clara de las ventas en el tiempo y permite realizar el seguimiento de las compras que realizan los clientes, facilitando así, información que puede ser explotada para realizar un mejor planeamiento de la producción y para el planteamiento de estrategias de marketing como por ejemplo de fidelización de los clientes.

Del análisis se deduce que la solución tecnológica ayuda a realizar un proceso de facturación y captación de pedidos más eficiente permitiendo así entregar los pedidos más rápidamente.

Por las razones expuestas se concluye que los objetivos de este trabajo fueron alcanzados satisfactoriamente.

## 6.2 Trabajos futuros

Como trabajos futuros se podrían implementar los siguientes módulos para la aplicación distribuida:

- Implementación de un módulo dentro del componente móvil que permita la funcionalidad de localización de clientes y pedidos a través de puntos marcados en mapas para que la entrega de los pedidos pueda realizar de manera más eficiente. Podría sugerirse el desarrollo de un algoritmo para el problema de ruteo de vehículos, con la finalidad minimizar el tiempo total en de transporte el que se realiza la entrega de los pedidos.
- Implementar la funcionalidad de seguimiento de la fuerza de ventas cuando son delegados para buscar nuevos clientes que permita verificar la localización exacta de los vendedores a través del sistema de posicionamiento global (GPS) del celular.
- Adecuar la aplicación distribuida para extender el alcance de esta aplicación poniendo en los diversos canales de distribución de aplicaciones móviles la app que permita realizar el registro de pedidos directamente por los clientes.
- Implementación de la funcionalidad para el componente cliente que permita el funcionamiento de modo offline y que al reestablecerse la conexión a internet se sincronice los datos hacia el servidor (esto en el caso de registro de pedidos y de clientes).

## Referencia Bibliográfica

### Libros

- Anderson, R. E., Hair Jr., J. F., & Bush, A. J. (1999). El proceso organizacional. (Mc Graw-Hill, Ed.), Administración de ventas (2a. ed.). Ciudad de México, México.
- Barry, D. K., & Dick, D. (2013). Web Services and Service-Oriented Architectures. En Elsevier Inc. (Ed.), Web Services, Service-Oriented Architectures, and Cloud Computing (2a ed., pp. 15–22). Waltham, MA, USA.
- Bondi, A. B. (2015). Foundations of Software and System Performance Engineering. (I. Pearson Education, Ed.). New York, NY, USA.
- Camden, R. K. (2016). Apache Cordova In Action. (Manning Publications Co., Ed.). Shelter Island, NY, USA.
- Camden, R., & Matthews, A. (2016). jQuery Mobile Web Development Essentials. (Packt Publishing Ltd., Ed.). Birmingham, West Midlands, England.
- Ceballos, F. J. (2008). Interfaces gráficas y aplicaciones para internet. (Rama, Ed.) (3a ed.). Madrid, España.
- De la Torre, C., Ramos, M. A., & Calvo, J. (2010). Estilos Arquitecturales. En Krasis PRESS (Ed.), Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0 (1a ed.).
- Di Nitto, E., Matthews, P., Petcu, D., & Solberg, A. (2017). Model-Driven Development and Operation of Multi-Cloud Applications. (Springer Open, Ed.).
- Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., ... Newling, T. (2004). A closer look at service-oriented architecture. En Patterns: Service-Oriented Architecture and Web Services (1a ed., pp. 24–26).
- Erl, T. (2016). Service Architecture, Service Composition Architecture, Service Inventory Architecture. En Prentice Hall (Ed.), Service-Oriented Architecture (2a ed., p. 81,89,95-97).
- Felke-Morris, T. (2017). Web development & Design foundations with HTML5. (I. Pearson Education, Ed.).
- Ferrer, J. (2012). Implantación de aplicaciones web. (RA-MA, Ed.). México, D.F.
- Forsyth, P. (2002). Control. En Capstone Publishing (Ed.), Sales Management (1a ed., pp. 56–57). Oxford, South East England, England.
- Gulabani, S. (2014). Developing RESTful Web Services with Jersey 2.0. (Packt Publishing Ltd., Ed.). Birmingham, West Midlands, England.
- Hossain, M. (2015). Cors In Action. Creating and consuming cross-origin APIs. (Manning Publications Co., Ed.). Shelter Island, NY, USA.

- Joines, S., Willenborg, R., & Hygh, K. (2002). Performance Analysis for Java Web Sites. (A. Wesley, Ed.) (1a ed.). Boston, MA, USA.
- Jonna, S. (2014). Learning PrimeFaces Extensions Development. (Packt Publishing Ltd., Ed.). Birmingham, West Midlands, England.
- Kalani, A., & Kalani, P. (2003). MCAD/MCSD : Developing XML Web Services and Server Components with Visual C# .NET and the Microsoft .NET Framework (1a ed.).
- Khare, T. (2012). Apache Tomcat 7 Essentials. (Packt Publishing, Ed.). Birmingham, West Midlands, England.
- Konstantas, D., Bourrières, J. P., Léonard, M., & Boudjlida, N. (2005). Interoperability of Enterprise Software and Applications. (Springer, Ed.).
- Krafzig, D., Banke, K., & Slama, D. (2004). What is a Service Oriented Architecture. En I. Pearson Education (Ed.), Enterprise SOA: Service-Oriented Architecture? (p. 57).
- Krakowiak, S. (2009). Middleware Architecture with Patterns and Frameworks. (Creative Commons license, Ed.).
- Laudon, K. C., & Laudon, J. P. (2012). Software empresarial. En Pearson Education (Ed.), Sistemas de información gerencial (12a ed., p. 338). Ciudad de México, México.
- Mauleón, M. (2003). Preparación de pedidos (Picking). En Ediciones Díaz de Santos (Ed.), Sistema de almacenaje y Picking (1a ed., pp. 217–218). Ciudad de Mexico, México.
- Mercado, S. (1997). Definición de administración de ventas. En Limusa (Ed.), Mercadotecnia programada: principios y aplicaciones para orientar la empresa hacia el mercado (2a ed., p. 343). Ciudad de Mexico, Mexico.
- Minter, D., Linwood, J., & Ottinger, J. (2010). Beginning Hibernate. (Apress, Ed.). New York, NY, USA.
- Mora, S. L. (2002). Arquitecturas cliente/servidor. En Club Universitario (Ed.), Programación de aplicaciones web: Historia, principios básicos y clientes web (pp. 40–44). Alicante, España.
- Peicevic, A. (2016). MySQL Introduction. (Geek University Press, Ed.).
- Richards, M. (2015). Layered Architecture. En I. O'Reilly Media (Ed.), Software Architecture Patterns. (1a ed., pp. 1–7). Sebastopol, CA, USA.
- Rotem-Gal-Oz, A. (2012). SOA Patterns. (Manning Publications Co., Ed.). Shelter Island, NY, USA.
- Santiago, R., Trbaldo, S., Kamijo, M., & Fernández, Á. (2015). Aplicaciones para m-learning. En Oceano (Ed.), Mobile Learning: Nuevas realidades en el aula (1a ed.). Pamplona, Navarra, España.
- Saternos, C. (2014). The Nature of the Web. En I. O'Reilly Media (Ed.), Client-Server Web Apps

- with JavaScript and Java: Rich, Scalable, and RESTful (pp. 6–7). Sebastopol, CA, USA.
- Stair, M. R., & Reynolds, G. W. (2017). ¿Qué es un sistema de información? En Cengage Learning (Ed.), Principios de sistemas de información (10a. ed., p. 10). Ciudad de México, México.
- Stanton, W. J., Etzel, M. J., & Walker, B. J. (2004). Proceso de las ventas personales. En McGraw-Hill (Ed.), Fundamentos de marketing (9a ed., p. 538). Ciudad de México, México.
- Visser, J. (2016). The four types of Software Maintenance. En I. O'Reilly Media (Ed.), Building Maintainable Software. Ten Guidelines for Future-Proof Code (1a ed., p. 2). Sebastopol, CA, USA.
- Wielenga, G. (2015). Beginning NetBeans IDE for Java Developers. (Apress, Ed.). New York, NY, USA.

### **Revistas**

- Barnett, B., Kirtland, M., & Ganapathy, M. (2003). An Architecture for Distributed Applications on the Internet: Overview of the Microsoft®.NET Framework. Journal of STEM Education : Innovations and Research, 7.
- IEEE. (1991). IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. IEEE Std 610, 174. <https://doi.org/10.1109/IEEESTD.1991.106963>
- Maffeis, S., & Olsen & Associates. (1997, diciembre). Client/Server Term Definition. Encyclopedia of Computer Science, D. Hemmendinger, A. Ralston, E. D. Reilly, eds. International Thomson Computer Publishing. Recuperado a partir de [http://www.maffeis.com/articles/research/client\\_server.pdf](http://www.maffeis.com/articles/research/client_server.pdf)

### **Congresos**

- Hau, T., Ebert, N., Hochstein, A., & Brenner, W. (2008). Where to Start with SOA: Criteria for Selecting SOA Projects. En Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008) (p. 4). IEEE. <https://doi.org/10.1109/HICSS.2008.501>

### **Reportes**

- González, Ó. (2000). Arquitecturas de sistemas de bases de datos. Universidad de Castilla de la Mancha.
- IBM. (2012). Native, web or hybrid. New York, NY, USA.
- Lionbridge. (2012). Mobile Web Apps vs. Mobile Native Apps: How to Make the Right Choice. Waltham, MA, USA.
- Oracle. (2014). The Java EE 7 Tutorial.
- Oracle. (2017a). JavaServer Faces Specification. Redwood Shores, CA, USA.
- Oracle. (2017b). JAX-RS: Java API for RESTful Web Services. Redwood City, CA, USA.



## **Patentes**

Taine, S.; Bandyopadhyay, S.; Sridharan, G.; Younis, S.; Ganesan, K. (2017). *U.S. Patent No. 9,838,375 B2*.

## **Tesis y proyectos de fin de carrera**

Armijos, A. E., & Pelaéz, V. E. (2008). Desarrollo de aplicaciones distribuidas online y smart client para dispositivos móviles utilizando .Net, que permitan acceder al sistema de administración estudiantil. Escuela Politécnica Nacional, Quito, Ecuador.

Balarezo, B. (2012). Desarrollo de un sistema de información de registro de pedidos para ventas usando dispositivos móviles. Pontifica Universidad Católica del Perú.

Narjiss Afellat, O. (2011). Proyecto de fin de carrera: Contratación y facturación electrónica en la cadena de valor. Universidad Carlos III de Madrid.

Víctor, M. (2011). Diseño e implementación de una aplicación distribuida de gestión de inventario para dispositivos móviles. Madrid, España: Universidad Carlos III de Madrid.

## **Páginas web**

Apache Software Foundation. (2010). Container Provided Filters. Recuperado 10 de febrero de 2018, a partir de <https://tomcat.apache.org/tomcat-7.0-doc/config/filter.html>

Artemisa. (2018). Presentación Artemisa. Recuperado a partir de <https://www.facebook.com/Artemisa-Lácteos-361077940732868/>

Asociación de marketing americano. (2014). Dictionary. Recuperado 2 de febrero de 2018, a partir de <https://www.ama.org/resources/Pages/Dictionary.aspx?dLetter=S#sales+management>

Bpmcapital. (2013). Administración de Ventas Móviles. Recuperado 4 de enero de 2018, a partir de <http://www.bpmcapital.com/AdmonVentasMoviles.html>

Creately. (2008). Creately. Recuperado a partir de <http://creately.com>

Digival. (2016). Diseño web responsable Vs. Aplicación para dispositivos móviles. Recuperado 21 de abril de 2018, a partir de <http://www.digival.info/disenio-web-responsable-vs-aplicacion-para-dispositivos-moviles/>

Gerencia. (2014). Soa Tecnología al servicio de los negocios. Recuperado 3 de enero de 2018, a partir de <http://www.emb.cl/gerencia/articulo.mvc?xid=1265>

Huber, B. (s. f.). Picking. Recuperado 24 de abril de 2018, a partir de <http://www.ilos.com.br/web/o-modelo-de-picking-ideal/>

Lucidchart. (2008). Lucidchart. Recuperado 3 de febrero de 2018, a partir de <https://www.lucidchart.com>

Microsoft Corporation. (2012). Infraestructura de servicios web XML. Recuperado 2 de noviembre de 2018, a partir de [https://msdn.microsoft.com/es-es/library/sd5s0c6d\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/sd5s0c6d(v=vs.100).aspx)

Netbeans Org. (2014). Getting Started with Cordova Applications. Recuperado 4 de febrero de 2018, a partir de <https://netbeans.org/kb/docs/webclient/cordova-gettingstarted.html>

Sevenminds. (2013a). Arquitectura y tecnología. Recuperado a partir de <http://sevenminds.com/pages/arquitectura-icloud.html>

Sevenminds. (2013b). Automatización Fuerza de Ventas. Recuperado 3 de enero de 2018, a partir de [www.sevenminds.com/pages/automatizacion-fuerza-ventas.html](http://www.sevenminds.com/pages/automatizacion-fuerza-ventas.html)

Sevenminds. (2013c). La Solución. Recuperado 3 de enero de 2018, a partir de <http://sevenminds.com/pages/solucion.html>

Sevenminds. (2016). Sevenminds - Captura de Datos. Recuperado 2 de enero de 2018, a partir de <https://play.google.com/store/apps/details?id=com.co.mysevenminds>

SUNAT. (2016a). Boleta de Venta. Recuperado 24 de abril de 2018, a partir de <http://orientacion.sunat.gob.pe/index.php/empresas-menu/comprobantes-de-pago-empresas/comprobantes-de-pago-fisicos-empresas/tipos-de-comprobantes-de-pago-fisicos-empresas/6587-02-boleta-de-venta>

SUNAT. (2016b). Comprobantes de Pago - Concepto. Recuperado 22 de abril de 2018, a partir de <http://orientacion.sunat.gob.pe/index.php/empresas-menu/comprobantes-de-pago-empresas/preguntas-frecuentes-comprobante-de-pago-empresas>

SUNAT. (2016c). Factura. Recuperado 23 de abril de 2018, a partir de <http://orientacion.sunat.gob.pe/index.php/empresas-menu/comprobantes-de-pago-empresas/comprobantes-de-pago-fisicos-empresas/tipos-de-comprobantes-de-pago-fisicos-empresas/6586-01-factura>

SUNAT. (2016d). Guías de Remisión - Comprobantes de Pago Físicos - Empresas. Recuperado 25 de abril de 2018, a partir de <http://orientacion.sunat.gob.pe/index.php/empresas-menu/comprobantes-de-pago-empresas/comprobantes-de-pago-fisicos-empresas/guias-de-remision-comprobantes-de-pago-fisicos-empresas/6600-01-definicion-y-aspectos-generales>


Vemobile. (2014). Conoce el producto. Recuperado 1 de febrero de 2018, a partir de <http://www.vemobile.mx/about.html>

DynamicReports. (s. f.). Recuperado 2 de febrero de 2018, a partir de <http://www.dynamicreports.org>

# **ANEXO A**

## **Validación del sistema**

A continuación se muestra la constancia de validación de la aplicación emitido por un especialista de la empresa Artemisa, realizando la validación de la funcionalidad así como de la importancia de este sistema para esta empresa:




### CONSTANCIA DE VALIDACIÓN

**NOMBRE DEL ESPECIALISTA:** PERCY, DÁVILA GONZÁLEZ  
**ESPECIALIDAD:** ADMINISTRADOR DE ARTEMISA - LÁCTEOS  
**DNI. N°** 40419531

Por medio de la presente hago constar que, he realizado la validación de la aplicación distribuida implementada por **LUIS IVÁN, MERA DÁVILA**, en el trabajo de TESIS Titulado: **"IMPLEMENTACIÓN DE UNA APLICACIÓN DISTRIBUIDA BASADA EN MÓVILES PARA EL APOYO A LA GESTIÓN DE VENTAS DE UNA EMPRESA DE PRODUCTOS LÁCTEOS"**. Esta validación consistió en la evaluación general del funcionamiento del sistema informático y la importancia de este sistema en la gestión de ventas de esta empresa.

Se expide el presente documento, de acuerdo a ley, para los fines que el interesado crea conveniente.

Chiclayo, 11 de Junio del 2018.



Producción y Comercialización de Productos Lácteos Naturales: Yogurt Natural Fritado (frutos secos y frutas frescas de estación), Queso Fresco, Mantequilla, Queso Mozzarella, Manjar Blanco y otros. Elaborado por Artemisa - Lácteos - RUC.: 10404195315 - Calle el Corregidor N° 111 Urb. Latina - José L. Ortiz - Chiclayo. Pedidos Delivery: (074) 250679 - 969593410 - 979356042 - f. artemisa - lácteos